

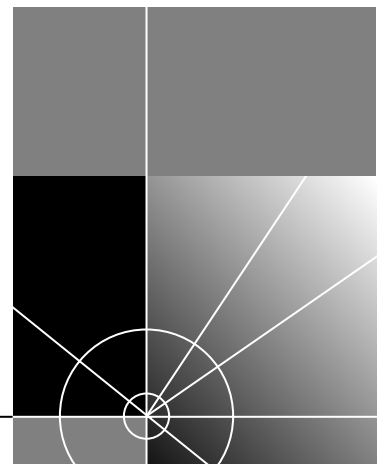


Using Enterprise OS Software

Version 11.4

<http://www.3com.com/>

Part No. 09-1861-000
Published January 2000



**3Com Corporation ■ 5400
Bayfront Plaza ■ Santa
Clara, California ■
95052-8145**

Copyright © **3Com Corporation, 2000**. All rights reserved. No part of this documentation may be reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without permission from 3Com Corporation.

3Com Corporation reserves the right to revise this documentation and to make changes in content from time to time without obligation on the part of 3Com Corporation to provide notification of such revision or change.

3Com Corporation provides this documentation without warranty of any kind, either implied or expressed, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. 3Com may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

UNITED STATES GOVERNMENT LEGENDS:

If you are a United States government agency, then this documentation and the software described herein are provided to you subject to the following restricted rights:

For units of the Department of Defense:

Restricted Rights Legend: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) for Restricted Rights in Technical Data and Computer Software Clause at 48 C.F.R. 52.227-7013. 3Com Corporation, 5400 Bayfront Plaza, Santa Clara, California 95052-8145.

For civilian agencies:

Restricted Rights Legend: Use, reproduction, or disclosure is subject to restrictions set forth in subparagraph (a) through (d) of the Commercial Computer Software - Restricted Rights Clause at 48 C.F.R. 52.227-19 and the limitations set forth in 3Com Corporation's standard commercial agreement for the software. Unpublished rights reserved under the copyright laws of the United States.

If there is any software on removable media described in this documentation, it is furnished under a license agreement included with the product as a separate document, in the hard copy documentation, or on the removable media in a directory file named LICENSE.TXT. If you are unable to locate a copy, please contact 3Com and a copy will be provided to you.

Unless otherwise indicated, 3Com registered trademarks are registered in the United States and may or may not be registered in other countries.

3Com, AccessBuilder, Boundary Routing, NETBuilder, and NETBuilder II, SuperStack, and OfficeConnect are registered trademarks of 3Com Corporation. Total Control and PathBuilder are trademarks of 3Com Corporation.

CompuServe is a registered trademark of CompuServe, Inc. IBM, AS/400, LAN Net Manager, OS/2, PS/2, and VTAM are registered trademarks of International Business Machines Corporation. Advanced Peer-to-Peer Networking and APPN are trademarks of International Business Machines Corporation. XNS is a trademark of Xerox Corporation. VAX, DEC, and DECnet are registered trademarks of Digital Equipment Corporation. TeleVideo is a registered trademark of TeleVideo Corporation. NetWare, Novell, and UNIX are registered trademarks of Novell, Inc. Banyan and VINES are registered trademarks of Banyan Systems. Telenet is a trademark of Telenet Communications Corporation. SPARCsystem is a trademark of SPARC International, Inc. licensed exclusively to Sun Microsystems, Inc. SunOS is a trademark of Sun Microsystems, Inc. Link level compression uses Stac LZS compression software, copyrighted by Stac Electronics, (© Stac Electronics, 1991-1995) and protected by one or more patents, including US patent 5,126,739. Stac and LZS compression are registered trademarks of Stac Electronics. AppleTalk and Macintosh are registered trademarks of Apple Corporation. Honeywell is a registered trademark of Honeywell Corporation.

Other brand and product names may be registered trademarks or trademarks of their respective holders.

The software contained in this product may contain encrypted product which may not be exported or transferred from the U.S. or Canada without an approved U.S. Department of Commerce export license.

Printed on recycled paper.

CONTENTS

ABOUT THIS GUIDE

Audience Description	60
Conventions	60
Year 2000 Compliance	61

CONFIGURING BASIC PORTS AND PATHS

Concepts	63
Paths	64
Ports	64
Port and Path Numbering on a NETBuilder II Bridge/Router	64
Port and Path Numbering on NETBuilder II Multiport Modules	65
Port and Path Numbering on a SuperStack II Bridge/Router	67
Port and Path Numbering on an OfficeConnect NETBuilder Bridge/Router	68
Configuring Multiple Paths to a Wide Area Port	69
Port and Path Numbering Issues for Built-in ISDN Interfaces	70
Configuring Local Area Interfaces	71
Configuring Wide Area Interfaces	72

CONFIGURING ADVANCED PORTS AND PATHS

Using Virtual Ports	77
Concept of Virtual Ports	77
Virtual Ports on SuperStack II Bridge/Routers	79
Virtual Ports and Different WAN Media	80
Virtual Ports over Frame Relay, ATM DXI, and X.25	80
Virtual Ports over ATM	81
Virtual Ports over PPP	82
Virtual Ports over SMDS	82
Parent Ports for X.25, PPP, Frame Relay, ATM and SMDS	82
Virtual Paths (WAN Extender only)	83
Configuring Virtual Ports	84
Virtual Ports for 802.1Q Virtual LANs	84
Using the Multiple Instance List	86
Instances	86
Groups	86
Defining a Group	87

Using a Group	87
Using Multiple Logical Networks	87
Concepts of Multiple Logical Networks	87
Configuring Multiple Logical Networks	91

CONFIGURING BRIDGING

Configuring Basic Bridging	93
Prerequisites	93
Transparent Bridging	93
Bridging over a Wide Area Network	93
Bridging over Multiple Logical Networks	94
Configuring for Bridging and Routing	96
Verifying the Configuration	97
Getting Statistics	99
Troubleshooting the Configuration	99
Customizing the Bridge	101
Per-Port Transparent Bridging	101
Adding or Deleting Static Entries	101
Bridge Security	101
Source Explicit Forwarding	102
Source Explicit Blocking	103
Destination Explicit Forwarding	104
Destination Explicit Blocking	105
Combined Source and Destination Security	106
Filters	107
Translation Bridging	108
Adding Functional-Address-to-Multicast-Address Mappings to the Default Table	109
Setting the Address Format	110
Optimizing Bridge Performance	110
How the Bridge Works	111
Transparent Bridging	111
IBM-Related Services	112
Token Ring Frame Copy Errors	113
Translation Bridging	113
OUI Packets	115
Maximum Transmission Unit	115
LLC Length and Packet Size	115
Address Mapping	115
Priority Mapping	116
Configuring Address Format	116
Protocol-Specific Issues	116
Spanning Tree Algorithm	117
How the Algorithm Works	118
Algorithm Requirements for Configuring the Network	118
How the Algorithm Creates a Loop-free Configuration	118
Using the Algorithm with Wide Area Bridges	121

Configuring the Spanning Tree Protocol over PPP	122
Spanning Tree Addressing	123
Modifying Spanning Tree Parameters	123
Reconfiguring the Topology	124
Load Sharing	124
Routing Tables	124
Learning and Filtering	125

CONFIGURING SOURCE ROUTE BRIDGING

Configuring a Basic Source Route Bridge	127
Prerequisites	127
Procedure	128
Configure Source Route Bridging over a Wide Area Network	130
Source Route Bridging over PPP	130
Source Route Bridging over Frame Relay, ATM, ATM DXI, and X.25	130
Source Route Bridging over SMDS	131
Source Route Bridging over ISDN	131
Verifying the Configuration	131
Getting Statistics	132
Troubleshooting the Configuration	133
Related Information	134
Customizing the Source Route Bridge	134
Enabling and Disabling Per-Port Source Route Bridging	135
Enabling and Disabling Per-Port Source Route Transparent Bridging	135
Configuring Source Route Transparent Bridging Gateway	136
Prerequisites	136
Procedure	136
Related Information	138
Connecting IBM Bridges to 3Com Token Ring Bridges	138
Procedure	138
Related Information	138
Configuring the Largest Frame Size	139
Configuring Passive Bridging	139
Procedure	139
Setting Up Spanning Tree	140
Configuring Parallel Bridges	141
Reducing Broadcast Traffic	141
Restricting Explorer Frame Propagation	142
Configuring Filters	142
Configuring Security	142
Configuring the Bridge/Router as an End System	142
Guidelines for Per-Port Route Discovery	142
Configuring Per-Port Route Discovery	143
Discovering Routes to an End System	144
Adding, Deleting, and Displaying Static Entries in the Routing Table	145
Aging Out Entries in the Routing Table	147
Changing the Token Access Priority	147

How the Source Route Bridge Works	147
Definitions	147
Source Route Bridging	147
Source Route Transparent Bridging	148
Source Route Transparent Bridging Gateway	148
IEEE 802.5 Token Ring Frame Format Overview	148
Source Route Transparent Bridging Gateway Concepts	150
Spanning Tree Considerations	150
Packet Handling between Domains	151
Frame and Address Conversion	152
Maximum Frame Size	154
Route Discovery Process	154
End System Source Routing	155
Routing Tables	156

CONFIGURING IP ROUTING

Configuring a Basic IP Router	157
Configuring for Local Area Networks and Point-to-Point Links	157
Prerequisites	157
Procedure	158
Related Information	159
Configuring for Wide Area Networks	159
Verifying the Configuration	160
Examining Network Devices	160
Checking with PING	160
Getting Statistics	162
Checking the Overall Status	162
Procedure	162
Related Information	163
Customizing the IP Router	163
Configuring UDP Broadcast Helper	163
Configuring Multiple IP Networks/Subnets	163
Related Information	164
Configuring Logical Networks over IP	164
Adding a Static IP Address	166
Configuring RIPv2 for Networks with Variable Length Subnet Masks	166
Using the Aggregate/Deaggregate Scheme	166
Using the Range Table Mask Scheme	168
Adding RIPv2 Compatibility	169
Adding Authentication	170
Adding Authentication Password	170
Configuring Static Routes	171
Procedure	171
Related Information	171
Configuring Packet Filtering	173
Procedure	173

Related Information	173
Configuring RIP Routing Policies	178
Prerequisites	179
Procedure	179
Migration to a RIPV2 Network	181
Configuring OSPF Routing Policies	181
Prerequisites	181
Procedure	182
Configuring OSPF Router Aggregation	183
Configuring IISIS Routing Policies	184
Prerequisites	184
Procedure	185
Using the IP Security Option	186
Configuring Interautonomous System Routing Using BGP	186
Configuring BGP Peers	186
Configuring a Default Route	187
Configuring BGP Route Aggregation	188
Importing Routes from IGP to a BGP Domain	189
Importing Routes from a BGP Domain to an IGP Domain	191
Configuring Network Number Policies	192
Configuring AS-Path Permit or Deny Policies	193
Configuring AS-Path Weight Policies	195
How the IP Router Works	196
Understanding IP Network Topology	197
Multipath Routing	199
Route Selection and Load Splitting	200
Route Selection Examples	201
Default Routes	201
Learning Routes within an Autonomous System	203
Learning Routes with RIP	203
Network Reachability	204
Solving the Slow Convergence Problem with Split Horizon	204
Solving the Slow Convergence Problem with Poison Reverse	206
User Configurations	206
Different States of RIP-Learned Routes	207
Learning Routes with OSPF	208
Configuring Integrated IS-IS for Dual IP and OSI Mode	215
Autonomous System Routing Using BGP	216
BGP Overview	216
External and Internal Peers	216
Peer-to-Peer Communication	217
Path Attributes	218
Path Selection	222
Policies	223
Route Aggregation	225

Address Resolution	225
Inverse ARP	226
Extended ARP	226
Other Global Router Configurations	226

CONFIGURING QUALITY OF SERVICE

Configuring IP Packet Classification Services	227
Configuring a Classifier List	227
Configuring an Address List	229
Configuring a Service List	229
Configuring IPQoS	229
POLicy Parameter	229
QoS Action Attributes	230
InVlanMap Per-port Parameter	232
OutVlanMap Per-port Parameter	232
TOSMap Per-port Parameter	233
CONTRol Per-port Parameter	233
About Policy-Based QoS Management	233
QoS Policies	234
IEEE 802.1P Prioritization	234
Class-Based Queueing Configuration	235
Configuring Per-Port CBQ Rate Limit	235
Configuring Class Based Queueing	237
ClassBasedQue Parameter	237
QueueStatistics Parameter	238
Examples	238
QueueCONTRol Parameter	239
IP Quality of Services Examples	239
Example 1:	240
Example 2	240
Example 3	240

CONFIGURING SYSTEM IP

Configuring System IP	243
Routing Issues	244
ARP Packets	245
NAT/Firewall Issues	245
VRRP Issues	245

CONFIGURING VIRTUAL PRIVATE NETWORKS

Using Tunnels	247
ISP to Central Site Tunneling	247
Remote User to Central Site Tunneling	248
Creating a VPN for Individual Remote Users	248
Example 1	248
Example 2	249
Creating a VPN for a Remote Office	250
On the Remote Office OfficeConnect Bridge/Router	251
On the Central Site PathBuilder Switch	251

CONFIGURING PUBLIC KEY INFRASTRUCTURE

Overview of PKI	253
Key-Pair Management	253
Certificate Enrollment	254
Certificate Installation and Storage	254
Certificate Usage and Validation	254
Certificate Revocation	255
Preparation for a Public Key Infrastructure	255
PKI Applications and Devices	256
Scope of the PKI	256
Certificate Authority Configuration	256
Device PKI Configuration	257
PKI Manager	258
Device PKI Dialog Facility	258
Initiating the Dialog Facility	258
PKI Dialog Facility Main Menu Options	258
Enrollment Key Management	259
Key-Pair Management	259
Certificate Requests	259
Certificate Fetch and Install	259
Certificate Display	260
CRL Cache Display	260
PKI File Display	260
PKI Command Line Interface	260
PKI Database Lock/Unlock	260
Key-pair Generation and Certificate Enrollment Commands	261
Unlock the PKI Database	261
Download and Install the CA Certificate Into Router	261
Generate an RSA Key-Pair	261
Set Up Parameters for the Certificate Request	262
Generate and Transfer the Request for the Router's Certificate	262
Acquire the Router's Certificate from the CA/RA	263
Add the Router's Certificate to the Local Certificate Database	263
Relock the PKI Database	263
Certificate/CRL Display	263

CRL Distribution Points	263
Remote Repository Default Addresses	263
Trust Policy Configuration	263
PKI Configuration Example	264

CONFIGURING PROTOCOL INDEPENDENT MULTICAST-SPARSE MODE

Configuring PIM-SM	267
Configuring PIM 1	267
Configuring PIM2	268
Procedure for PIM3	269
Procedure for PIM4	269
How PIM-SM Works	270
Multicast Routing Mechanisms	270
Rendezvous Points	270
Bootstrap Router	270
Joining a Group	271
Sending Data to a Group	271
Switching from a Shared Tree to a Shortest Path Tree	271
Leaving a Group	271
PIM Packet Formats	271

CONFIGURING RSVP

What Is RSVP?	273
RSVP Configuration Example	274
RSVP Proxy Sender and Receiver	274
Proxy Sender: Unicast Destination and One Sender Port	274
Proxy Receiver: Unicast Destination and One Sender Port	275
Proxy Sender: Multicast Destination with a Range of Sender Ports	275
Proxy Receiver: Multicast Destination with a Range of Sender Ports	275
Sample RSVP Configuration with L2TP Tunnel	276

CONFIGURING DHCP

Configuring DHCP	279
Procedure	279
Configuring Address Pools	280
Enabling DHCP Service	280
Configuring Address Pools	280
AddressPool	281
DefAddressPool	281
Procedure	281
Procedure	282
Configuring the DHCP Options	282
Procedure	282
Creating a New DHCP Profile	283
Procedures	283

Configuring DHCP Profiles	284
Logging to the Console	286

CONFIGURING L2TUNNEL CONNECTIONS

Configuring a NETBuilder Bridge/Router as a Tunnel Terminator (PP)	289
Configuring a NETBuilder Bridge/Router as a Tunnel Initiator/Terminator (Router-to-Router)	292
Configuring Virtual Leased Line over PPTP/L2TP	292
Configuring Tunnel Switching	295
Configure Tunnel Switching Using the Radius Server	296
Tunnel Security	296

CONFIGURING TUNNEL ROUTE SHORT CUTS

Configuring R1	300
Configuring R2	301
Configuring R3	302
Address and Routing Table Displays	303
Router Kicks Off an NHRP Resolution Request	304
Router Receives an NHRP Resolution Reply	304
Router Receives an NHRP Resolution Request	304

CONFIGURING IPV6 ROUTING

Configuring a Basic IPv6 Router	307
Configuring for Local Area Networks	307
Prerequisites	307
Procedure	307
Related Information	308
Configuring for Wide Area Network Connectivity	308
Verifying the Configuration	308
Examining Network Devices	308
Getting Statistics	309
Checking the Overall Status	309
Checking with Ping6, TraceRoute6, and Telnet6	309
Customizing the IPv6 Router	310
Configuring the Internal Port	310
Configuring Multiple IPv6 Subnets	310
Configuring Neighbor Discovery	311
Configuring Static Routes	311
Procedure	311
Configuring RIPNG Routing Policies	311
Prerequisites	311
Procedure	312
Configure IPv6 Routing Over ATM using PVCs	313
Configuring RIPNG Route Aggregation	313

How the IPv6 Router Works	314
Understanding IPv6 Network Topology	314
Multipath Routing	314
Route Selection and Load Splitting	315
Default Routes	315
Learning Routes with RIPNG	316
Network Reachability	316
Solving the Slow Convergence Problem with Split Horizon	317
Solving the Slow Convergence Problem with Poison Reverse	317
Different States of RIPNG-Learned Routes	317
IPv6 Transition Support	318
Automatic Tunnel	318
Point-to-Point Tunnel	318

CONFIGURING NETWORK ADDRESS TRANSLATION

Configuring Network Address Translation	319
Prerequisites	319
Enabling NAT Ports	319
Defining the Address Mapping	319
Defining TCP/UDP Port Mapping	320
Logging Messages	320
Session Information	321
Translation failure actions	321
Adding a Dynamic Address Map	321
How NAT Works	321
When to Use NAT	322
Guidelines	322
Basic NAT Operation	322
Specifying Direction	323
Address Mapping	324
NAT Proxy ARP	325
Using a Mask	325
IPCPAddress Mapping	325
TCP/UDP Port Mapping	327
Mapping an address and TCP/UDP port to another address and TCP/UDP port allows more control of the type of traffic NAT translates.	327
NAT Scenarios	327
Private Address Space	327
Load Sharing	330
Address Migration	330
Address Redirection	331
IPSEC and NAT	332
AH Packet Type	332
ESP Packet Type	333
Limitation on One-to-One Mapping	333

CONFIGURING IP MULTICAST ROUTING

Configuring a Basic Multicast Router	335
Configuring for Local Area Networks and Point-to-Point Links	335
Prerequisites	335
Procedure	335
Configuring for Wide Area Networks	337
Verifying the Configuration	337
Checking the Overall Status	337
Getting Statistics	338
Troubleshooting the DVMRP Configuration	338
Customizing the Multicast Router	339
Controlling Local Group Membership Queries	340
Configuring an IGMP Proxy Agent and IGMP Version	340
Adjusting the Multicast Datagram Threshold	341
Configuring Multicasting over SMDS	342
Using the DVMRP Protocol	343
Configuring a DVMRP Multicast Tunnel	343
Configuring DVMRP Scoping	344
Configuring DVMRP Multicasting over Frame Relay	345
Configuring DVMRP Multicasting over X.25	346
Configuring a DVMRP Metric	347
Controlling the DVMRP Rate Limit for Multicast Traffic	347
Configuring DVMRP Route Aggregation	348
Controlling the Routing Table	349
Controlling the Forwarding Table	350
Using the MOSPF Protocol	351
Configuring Interarea Multicasting	352
Configuring MOSPF Routing Policies	352
Configuring MOSPF Forwarding Policies	353
Configuring a Multicast Border Router	355
How the IP Multicast Router Works	355
MBONE Connectivity with Multicasting	356
Multicast Addresses	357
Internet Group Management Protocol	358
Distance Vector Multicast Routing Protocol	358
Routing Table	359
Forwarding Table	360
Multicast Open Shortest Path First Protocol	360
Learning Group Membership	361
Shortest Path First Tree	361
Forwarding Cache	362
Interarea Multicasting	362
Interautonomous System Multicasting	363
Multicast Routing Terms	363

CONFIGURING THE VIRTUAL ROUTER REDUNDANCY PROTOCOL

Configuring VRRP	367
Supported Media and Protocols	367
Prerequisites	367
Enable the Owner Router	367
Enable the Backup Routers	368
Setting Priorities for Multiple Backup Routers	368
Setting the Hold Time	369
Setting the Priority	370
Enabling VRRP	370
Ping/Telnet Virtual Router IP (VIP)	370
Disabling/Deleting VRRP	371
Customizing VRRP	371
Setting the Advertisement Interval	371
How VRRP Works	371
MAC Address	372
Scenarios	372
Gateway to a WAN	372
Connecting Two LANs	372
Load Sharing with Redundancy	373
DLSw Resilient Tunnels	374
VRRP for Token Ring	376
Functional Address Mode	376
Unicast Address Mode	376

CONFIGURING THE ROUTER DISCOVERY PROTOCOL

Setting Up RDP	377
Prerequisites	377
Procedures	377
Defining Participating Routers	378
Configuring the Timers	378
Enabling and Disabling RDP	379
Discovering Neighboring RDP Routers	379
Verifying the RDP Configuration	380
Troubleshooting the RDP Configuration	380
How RDP Works	380
RDP Features	381
Other Timer Considerations	381
RDP Terms	381

CONFIGURING THE REMOTE POLLING PROTOCOL

- Configuring REMP 383
 - Adding Static Targets 383
 - Dynamic Targets 384
 - Configuring Priority 385

CONFIGURING UDP BROADCAST HELPER

- Configuring UDP Broadcast Helper 387
 - Prerequisites 388
 - Procedure 388
- Relaying BOOTP and DHCP Traffic 391
 - Prerequisites 391
 - Procedure 391
- Verifying the Configuration 393
 - Checking Parameter Settings 393
 - Getting Statistics 393
- Customizing the Configuration for BOOTP 393
 - Limiting the Number of Hops 393
 - Prerequisites 393
 - Procedure 394
 - Determining Order of Booting 394
 - Prerequisites 394
 - Procedure 395
- How UDP Broadcast Helper Works 395
 - BOOTP and DHCP Protocols 396

BUILDING INTERNET FIREWALLS

- Setting Up an Internet Firewall 397
 - Prerequisites 397
 - Defining Your Firewall Stance 398
 - Continuing Routing Functions 399
 - Configuring OAM Procedures 399
 - Configuring Telnet 399
 - Configuring TFTP 400
 - Configuring ICMP (Ping) 400
 - Configuring SNMP 400
 - Configuring FTP 400
 - Verifying the Configuration 400
 - Checking the Overall Status 401
 - Blocking Unwanted Traffic 402
- Configuring a Firewall for IP Security Protocol Encrypted Packets 404
 - Specifying Packet Handling 404
 - Support for IP Security 404
 - User-defined Services 405
 - Defining a Service 405

Configuring a Firewall Using a User Defined Service Name	406
Differences Between a Predefined Service Filter and a User Defined Service Based Filter	406
Using IP Addresses Grouping	407
Using the RealPlayer Predefined Service	407
Firewall Configuration	408
Additional Predefined Services	409
Enabling Security	410
Managing Filters	410
Filter Rule Syntax	411
Creating Filters Using Filter Rules	411
Defining a Filter Using the ADD Filter Command	411
Creating Filters Using An Off-line Editor	411
Displaying Filters	412
Deleting Filters	412
Assigning Filters to Interfaces	412
Activating and Deactivating Filters	413
Firewall Filters versus IP Filters	413
Filters — Firewall Execution Order	413
Setting Up System Logs	414
Specifying Log Content	414
Log Description	414
Firewall Log Examples	416
How a Firewall Works	417
Packet-Filtering Routers	417
Benefits of Packet-Filtering Routers	418
Firewall Filter Types	418
Service-Independent Filters	418
Predefined (Service-Dependent) Filters	418
Dynamic “ Window Management ” for FTP	419
Generic Filters	420
Firewall Terms	420

CONFIGURING THE ACCESS CONTROL SERVICE USING RADIUS

Configuring AC	423
----------------	-----

CONFIGURING THE LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL FOR FIREWALL CONFIGURATION

Configuring LDAP	425
Setting Up LDAP	425
Example	425
Verifying the LDAP Configuration	426
Adding Attributes and ObjectClasses on the Netscape Directory Server	426
Adding Attributes	426
Creating an Object Class	427

Populating Database on LDAP Server	427
Creating Database in LDAP Data Interchange Format (LDIF)	428
General LDIF Rules	429
How LDAP Works	430
LDAP Terms	430
3Com Proprietary Attributes and Object Classes Definitions	431

CONFIGURING REMOTE ACCESS SERVICES

Configuring Remote Access	433
Configuring Distributed Remote Access with a NETBuilder Tunnel Terminator	433
Prerequisites	433
Procedure	434
Configuring Distributed Remote Access for a NetWare Network	436
Windows 95 with Dialup Networking 1.2 or Windows 98 or Windows NT 4.0 Client Options	437
Internet Based Remote Access	437
OfficeConnect NETBuilder with NAT at the Remote Office	438
438	
Configuring NCPs	438
Example	439
RADIUS Framed Net Mask and Framed Route Attributes	439
Example	439
RADIUS NAS-Port-Type Attribute	440
Specifying Log Content	441

IP SECURITY OPTIONS

Configuring IP Security Parameters for End Systems	443
Prerequisites	443
Procedure	444
Configuring IP Security Options for IP Routers	444
Prerequisites	445
Procedures	445
Port 1 Configuration	446
Port 2 Configuration	446
Port 3 Configuration	447
Port 4 Configuration	448
Enabling IP Security Processing	449
Configuring Extended Security Option Labels	449
Verifying IP Security Options	450
ICMP Error Messages	450
Preventing Security Attacks on IP Routers	450
How IP Spoofing Works	450
Hijacking Tool	451
Preventing Attacks	451
Secure Configuration Solutions	452

Noncontiguous IP Networks	452
Subnets on the Internal Network	453
Multiple Contiguous IP Networks	454
Alternative Two-Router Configurations	454
Firewall Configurations	455
IP Security Terms	455

CONFIGURING IPSEC

Configuring IPsec	457
Creating Manual Policies	457
Configuring Manual Security Policies	458
Creating Key Sets	458
Configuring Manual Key Information	459
Configuring IPsec with Manual Policy	460
Configuring Dynamic-Key Security Policies	460
Selector Lists	461
Transform Lists	461
IKEProfile	461
PreSharedKey	461
DynamicPolicy	461
Customized Security Associations	461
Enabling IPsec	462
How IPsec Works	462
Policies	462
Encapsulation Security Payload	462
Authentication Header	463
IP Payload Compression	463
Sample Configurations	463
Creating a Manual Security Policy in Transport Mode	463
Manual Key: Setting up a VPN PPTP Tunnel	464
Router 1	465
Router 2	466
Establishing the Dialup Tunnel	466
Manual Key: Creating a Fully Meshed Topology Between Three Routers	466
Router 1	467
Router 2	468
Router 3	468
Dynamic Key: Creating a Fully Meshed Topology Between Three Routers	469
Router 1	469
Router 2	470
Router 3	471
Dynamic Key: Hub and Spoke Topology Between Three Routers	472
Router 1, Router 2, and Router 3	472
Dynamic Key: Hub and Spoke Topology Between Three Routers (Intranet/Extranet)	473
Spoke Router 1	475
Spoke Router 2 (Intranet)	476

Spoke Router 3	477
Extranet Router 4	478

CONFIGURING APPN INTERMEDIATE SESSION ROUTING

Setting Up a Basic APPN Router	481
Setting Up Your System as a Network Node	483
Prerequisites	483
Procedure	484
Defining Links to Other Network Nodes	486
Procedure	487
Configuring Dependent LU Support	490
Defining the Default DLUs and Backup DLUs	491
Defining Upstream Links for Path to DLUs	492
Defining Downstream Links to Nodes with Dependent LUs	492
Using VTAM Program Temporary Fixes	493
Enabling the Network Node and Activating Links	493
Dynamic Configuration Options	494
Configuring the APPN Router for Wide Area Networks	496
Verifying the APPN Router Configuration	496
Troubleshooting the APPN Router	497
Customizing the APPN Router	498
Defining Links to End Nodes	499
Defining Links to Unknown Node Types	500
Defining Entries in the Network Node's Directory	500
Preconfiguring LEN End Node LUs	501
Deleting LEN End Node LUs	503
Adding Entries	503
Deleting Entries	504
Configuring Parallel Transmission Groups	505
Configuring Parallel TGs on the Network Node	506
CP-CP Sessions on Parallel TGs	508
Parallel TGs and Source Route Dual-TIC Topologies	508
Configuring DLSw Between Network Nodes	508
Configuring APPN for Boundary Routing	510
Configuring APPN Connection Networks	511
Using Connection Networks to Scale Larger Networks	511
Configuring Links to Connection Networks	513
Using Connection Networks in Boundary Routing Environments	513
Operating the Network Node	515
Disabling the Network Node	515
Deleting Links to Adjacent Nodes	516
Activating and Deactivating APPN Ports and Links	516
Activating and Deactivating Ports	516
Activating and Deactivating Links	517
Pinging to APPN Network Resources	518
Displaying APPN Information	518
APPN Directory Information	518

Network Topology Information	519
Adjacent Link Station Information	520
Current Status of APPN Ports	520
Active APPN Connections	520
Current Status of Link Stations	521
Current Status of Adjacent Nodes	521
Intermediate Session Routing Information	521
How APPN ISR Routing Works	523
APPN Node Types	523
Network Nodes	524
End Nodes	525
Low-Entry Networking End Nodes	525
Differences Between Network Nodes and End Nodes	525
Network Node Role	526
How the Network Node Directory Learns About Local End Node LU Resources	527
How the Network Node Discovers the Location of Destination LUs	528
Additional Information	530
Fully Qualified and Not Fully Qualified CP Name Formats	530
MAC Address Format Options for APPN	531
Setting the Maximum BTU Size	532
APPN Terms	533
IBM APPN References	534

APPN HIGH PERFORMANCE ROUTING

Configuring the Network Node to Perform HPR	535
Prerequisites	535
Procedure	536
Configuring HPR Subnets within ISR Networks	538
Using HPR with Boundary Routing Environments	539
Operating the HPR Network Node	540
Setting RTP Connection Timers	540
Displaying RTP Connections	540
Initiating a Nondisruptive Path Switch	540
How HPR Works	541
HPR Node Types	541
IBM Devices Supporting HPR	542
Automatic Network Routing	542
Rapid Transport Protocol	543
RTP Connections	543
Nondisruptive Path Switching	544
Adaptive Rate Pacing	546
Comparison of ISR and HPR Functions	547

CONFIGURING APPN CLASS OF SERVICE

Default SNA Class of Service Modes	549
Creating Customized Class of Service Tables	550
Mapping Class of Service Names to Mode Names	551
Displaying Class of Service Information	551
Deleting Class of Service Information	551
How Class of Service Calculates Routes	552
Step 1: Determining Node Weights Along a Path	553
Step 2: Determining TG Weights Along a Path	555
Step 3: Calculating the Total Weight for Each Path	557
Default Class of Service Tables	558
Default Node Table	558
Default TG Tables	558

CONFIGURING IPX ROUTING

Setting Up a Basic IPX Router	563
Configuring for Local Area Networks and Point-to-Point Links	563
Prerequisites	563
Procedure	563
Configuring Secondary Networks with Different Header Formats	564
Configuring for Wide Area Networks	566
Configuring IPXWAN over PPP	567
Prerequisites	567
Procedure	567
Configuring for NLSP	569
Prerequisites	569
Procedure	569
Verifying the Configuration	570
Getting Statistics	572
Troubleshooting the Configuration	572
Customizing the IPX Router	574
Controlling NRIP and SAP Advertisements	575
Enabling and Disabling Dynamic Learning and NRIP Updates	575
Enabling Triggered NRIP Updates	575
Using Poison Reverse or No Poison Reverse	576
Controlling NRIP and SAP Updates	576
Controlling Route and Service Aging	577
Flushing Dynamic Routes and Server Table Entries	577
Flushing Dynamically Learned WAN Neighbors	578
Built-in IPX Masks	578
User-defined IPX Masks	578
Adding and Deleting Static Routes	579
Prerequisites	580
Procedure	580
Configuring a Static Default Route	582
Procedure	582

Configuring a Default Metric	583
Adding and Deleting Static Servers	584
Configuring Neighbor Policy	584
Writing NRIP and SAP Policies for IPX	585
NETBuilder II Examples	586
SuperStack II Examples	588
Configuring Other Policy Settings	588
Configuring IPX Spoofing over a DOD Link	589
NCP Spoofing over a DOD Link	589
NCP Keep Alive Mechanism	590
Supported Configurations	591
SPX1 Spoofing Lite over a DOD Link	593
Supported Configurations	594
How the IPX Router Works	595
IPX Router Features	596
Local and Wide Area Network Configuration	596
Routing Tables	597
Default Routes	599
Effect on NRIP	599
Effect on NLSP	599
Effect on SAP	599
Routing Selection	599
Learning Routes and Service Information	600
Server Tables	601
Network Reachability	601
Solving the Slow Convergence Problem with Split Horizon	601
Solving the Slow Convergence Problem with Poison Reverse	603
Route, Service, and Neighbor Policies	603
Policy Control	604
Route Receive Policy	605
Route Advertisement Policy	605
Service Receive Policy	606
Service Advertisement Policy	607
Neighbor Policy	607
Novell Service Types	608
NLSP Routing	609
Hierarchical Routing	609
Area Addressing	610
IPX Routing Terms	611

CONFIGURING APPLE TALK ROUTING

- Setting Up a Basic AppleTalk Router 613
 - Prerequisites 613
 - Creating a Router Plan 613
 - Procedures 614
 - Configuring for Local Area Networks 614
 - Configuring for Wide Area Networks 615
 - Related Information 616
- Verifying the Configuration 617
 - Getting Statistics 618
 - Troubleshooting the Configuration 618
- Customizing the AppleTalk Router 619
 - Setting Up Multiple Seed Routers 619
 - Procedure 619
 - Related Information 620
 - Setting Up AppleTalk Routing over a Non-AppleTalk Data Link 620
 - Related Information 620
 - Changing Frequency of Routing Table Route Propagation 621
 - Procedure 621
 - Related Information 621
 - Setting Up Filters 622
 - Setting Up Network Number-Based Filtering 622
 - Setting Up Entity Filters 624
 - Setting Up Zone Advertisement Filtering 626
 - Procedure 627
 - Procedure 627
- Changing a Zone List 628
- How the AppleTalk Router Works 628
 - Network Entities 630
 - Port Startup Operations 633
 - Network AppleTalk Operations 634
 - Split Horizon 635
 - AppleTalk over PPP 635
 - Filtering on Frame Relay Ports 635
 - Routing Table 635

CONFIGURING DECNET ROUTING

Setting Up a Basic DECnet Router	637
Configuring for Local Area Networks and Point-to-Point Links	637
Prerequisites	637
Procedure	637
Configuring for Wide Area Networks	639
Verifying the Configuration	639
Getting Statistics	639
Troubleshooting the Configuration	640
Customizing the Configuration	641
Controlling Routing Information	641
Procedure	641
Related Information	641
Setting the Priority	642
Setting the Cost	642
Enabling and Disabling Triggered Routing Updates	642
Setting the Routing Time	642
Setting the Hello Messages Time	643
Procedure	643
Related Information	643
How the DECnet Router Works	643
DECnet Network	643
Routing Tables	644
Learning Routes	646
Network Reachability and Split Horizon	646
Cost-effective Routing	647
Routing Phase IV Traffic over DOD Lines	647
Address Translation Gateway Support	647
Internetwork Routing Support	647
Address Translation	647
Address Translation Configuration Example	648
Internetwork Boundary Routing	649
Phase IV to Phase V Transition Support	650
Phase IV to Phase V Translation	650
DECnet Area to Pseudo Areas Translation	651
Pseudo Area Configuration	652
Phase IV to Phase V Transition Configuration Example	653
DECnet Phase V and Phase IV Terms	654

CONFIGURING OSI ROUTING

Setting Up a Basic OSI Router	657
Configuring for Local Area Networks and Point-to-Point Protocol Links	657
Prerequisites	657
Procedures	657
Configuring for Wide Area Networks	659
Verifying the Configuration	659
Checking Packet- Forwarding Process	660
Getting Statistics	662
Troubleshooting the Configuration	662
Incomplete Level 2 Backbone	662
Partitioned Area	663
Multiple Area Addresses	664
Mismatched Passwords	664
Customizing the OSI Router	664
How the OSI Router Works	665
OSI Network Topology	665
Area Addresses	666
ID and Selector Values	667
Network Entity Title	667
Areas	668
Level 1 Routing	668
Level 1 Routing Table	669
Level 2 Routing	670
Level 2 Routing Table	672
Transit and Leaf Areas	672
Metrics and Route Selection	673
Multipath Routing and Load Splitting	674
End System Table	674
Intermediate System Table	674
User Configurations	674
Setting Up Interdomain Routing	676
Prerequisites	676
Procedure	676
Related Information	677
Integrated IS-IS for IP and Dual IP/OSI Mode	681

CONFIGURING VINES ROUTING

Setting Up a Basic VINES Router	683
Configuring for Local Area Networks and Point-to-Point Protocol Links	683
Prerequisites	683
Procedure	683
Configuring for Wide Area Networks	684
Verifying the Configuration	685
Verifying Procedure	685
Getting Statistics	685
Checking Reachability	685
Troubleshooting the Configuration	686
Procedure	686
Customizing the VINES Router	687
How the VINES Router Works	687
Routing Tables	688
VINES Routing Table	688
VINES Neighbor Table	690
Routing Selection	691
Deleting Routes	691
Learning Routes	691
Network Reachability, Split Horizon, and UpdateTime	691
Banyan VINES Client/Server Support	692

CONFIGURING XNS ROUTING

Setting Up a Basic XNS Router	693
Configuring for Local Area Networks and Point-to-Point Protocol Links	693
Prerequisites	693
Procedure	693
Configuring for Wide Area Networks	694
Verifying the Configuration	694
Getting Statistics	696
Troubleshooting the Configuration	696
Customizing the XNS Router	697
Local and Wide Area Network Configuration	697
Defining Routes	698
Static Routes	698
Dynamic Routes	698
Enhancing the Performance of the XNS Router	698
Configuring for RIP Updates	698
Configuring for Error Checking	701
How the XNS Router Works	701
Learning Routes	701
Displaying Routing Information	701
Deleting Routes	702
Network Reachability and Split Horizon	702

CONFIGURING THE LLC2 DATA LINK INTERFACE

Configuring LLC2 Data Link Interface	705
Displaying LLC2 Information	706
Configuring LLC2 with Other Services	707

CONFIGURING SNA NETWORKS USING QLLC TO LLC2 CONVERSION

Setting Up QLLC to LLC2 Conversion	709
Prerequisites	709
Procedure	709
How QLLC to LLC2 Conversion Works	710
QLLC Acronyms	712
Limitations	712

CONFIGURING SYNCHRONOUS DATA LINK CONTROL CONNECTIVITY

- Connection Methods 713
- Configuring the Router for SDLC 714
 - Prerequisites 714
 - Procedure 715
 - Configuring the SDLC Port and Path Attributes 715
 - Configuring LLC2 and Bridging Characteristics 716
 - Configuring the SDLC Protocol Characteristics 716
 - Configuring the SDLC Protocol Timing Parameters 717
- Configuring the CU Devices on the Link 717
 - Prerequisites 717
 - Procedure 717
- Verifying the Configuration 719
- Using Frame Relay Access 720
- APPN over SDLC 720
- How SDLC Conversion Works 721
 - Address Mapping 722
 - Session Initiation 724

CONFIGURING SDLC AND HDLC TUNNELING FOR SNA NETWORKS

- Configuring SDLC and HDLC Tunneling 725
 - Prerequisites 725
 - Procedure 725
 - Configuring Router A 726
 - Configuring Router B 728
 - Verifying the Configuration 728
 - Displaying Circuits 729
- How SDLC and HDLC Tunneling Works 729

CONFIGURING DATA LINK SWITCHING FOR SNA AND NETBIOS NETWORKS

- Configuring for SNA 731
 - Prerequisites 731
 - Procedure 732
- Configuring for NetBIOS 734
 - Prerequisites 734
 - Procedure 735
- Verifying the Configuration 737
 - Displaying Connections 737
 - Displaying Circuits 737
 - Displaying LLC Sessions 738
 - Displaying Cache 738
 - Displaying the DLSw Activity Log 738
 - Displaying the DLSw End-Station Topology 739
- Customizing the Configurations 741

Defining a Non-Secure Host Configuration	741
Prerequisites	742
Procedure	742
Setting Up DLSw Security Access Filters	743
Setting Up Filters for SNA Traffic	743
Setting Up Filters for NetBIOS Traffic	744
Disabling Data Link Switched Connections	744
Configuring Statically Defined Media Addresses	745
Configuring Statically Defined NetBIOS Names	745
Prioritizing DLSw Traffic	745
How Prioritization and Bandwidth Allocation Work	745
Configuring Bandwidth Allocations and Priorities	747
Prerequisites	747
Procedure	747
Examples of Other Commands	748
Prioritizing DLSw Packets	749
Circuit Balancing	749
How Circuit Balancing Works	749
Configuring Circuit Balancing	750
Prerequisites	750
Procedure	751
Examples of Other Circuit Balancing Commands	751
Configuring Local Switching and Port Groups	751
Using Local Switching to Translate Different DLC Traffic Types	751
Configuring Port Groups for Funneling Many Remote Connections Into Fewer DLSw Connections	753
Network Design Issues for Port Grouping	756
Configuring DLSw for Dual-TIC Topologies	758
Converting SNA Alerts to SNMP Traps	759
How SNA-Alerts-To-Traps Works	759
Configuring SnaAlertsToTraps	759
Enabling DLSw Loop Detection	760
Initiating Loop Detection	761
How Data Link Switching Works	761
Media Addressing and NetBIOS Name Caching	762
DLSw Configuration and STP	762
Data Link Switching Terms	763

CONFIGURING MULTICAST DATA LINK SWITCHING FOR NETBIOS AND SNA NETWORKS

- Configuring Multicast DLSw 765
 - Configuring DLSw Multicast for NetBIOS Mesh Environments 766
 - Prerequisites 766
 - Configuring Multicast DLSw for SNA Client and Server Environments 767
 - Prerequisites 767
- Customizing the DLSw Multicast Configuration 769
 - Tuning DLSw Multicast Parameters 769
 - Restoring the Default Multicast Address 769
 - Disabling DLSw Multicast 770

CONFIGURING FRAME RELAY ACCESS DEVICE SUPPORT FOR SNA

- Configuring the NETBuilder as a FRAD Node 771
 - Configuring FRAD for LAN-Attached End Stations 771
 - Configuring the FRAD Node for a BAN-Attached End Station 771
 - Prerequisites 772
 - Configuring the FRAD Node for a LAN-Attached End Station Using BNN 773
 - Prerequisites 773
 - Configuring FRAD for SDLC-Attached End Stations 774
 - Configuring the FRAD Node for an SDLC-Attached End Station Using BAN 774
 - Prerequisites 774
 - Procedure 774
 - Configuring the FRAD Node for an SDLC-Attached End Station Using BNN 775
 - Prerequisites 775
 - Procedure 776
 - Deleting Frame Relay Address Mappings 777
 - Displaying Frame Relay Address Mappings 777
- How the Frame Relay Access Device Works 777
 - BNN Configuration 778
 - BAN Configuration 780

CONFIGURING LAN ADDRESS ADMINISTRATION

- Assigning a MAC Address to a Physical Path 781
- Assigning a MAC Address to a Main Processor Module Interface 783
- Using Duplicate MAC Addresses for SNA Load Balancing 783
- Using LAA with DECnet 784

CONFIGURING NETVIEW SERVICE POINT

- Configuring NetView Service Point 787
 - Activating and Deactivating SSCP Link Stations 789
 - Activating and Deactivating All SSCP-PU Sessions 789
 - Checking LU Status 790

CONFIGURING BINARY SYNCHRONOUS COMMUNICATIONS CONNECTIVITY

- Configuring BSC Pass-Through 791
 - Prerequisites 791
 - Remote Site Configuration 792
 - Central Site Configuration 793
 - Baud Rate and Line Speed Considerations 794
 - Modifying Existing BSC CU Definitions 795
 - BSC Configuration Examples 795
 - Example 1: CU At Single Remote Site 795
 - Example 2: Multiple CUs On One Port at a Remote Site 796
 - Example 3: CUs at Multiple Remote Sites 797
- Configuring BSC Conversion 799
 - Prerequisites 799
 - BSC Conversion Configuration 800
 - BSC Conversion Examples 801
 - Example 1: Single CU With Multiple Devices At Single Remote Site 801
 - Example 2: Multiple ATM CUs On One Port at a Remote Site 802

CONFIGURING POLLED ASYNCH CONNECTIVITY

- Configuring Asynch Tunnels on Both Central and Remote Sites 805
 - Prerequisites 805
 - General Asynch Port and Path Configuration 806
 - Asynch Port Configuration 807
 - Asynch CU Configuration 809
 - Asynch Tunneling Configuration Examples 811
 - Example 1: Single Asynch Devices at the Remote Sites 811
 - Example 2: Multiple Asynch Devices at Remote Sites 813

CONFIGURING BOUNDARY ROUTING SYSTEM ARCHITECTURE

Configuring Basic Boundary Routing	817
Prerequisites	817
Configuring for PPP	817
Configuring for Frame Relay	822
Configuring for X.25	828
Verifying the Configuration	832
Troubleshooting the Configuration	834
Customizing Boundary Routing	836
Configuring Dial-Related Enhancements	836
Configuring Dual PVCs in a Boundary Routing Environment	836
Configuring Dual PVCs on the Central Node	837
Verifying the Dual PVC Configuration	840
Configuring Network Resiliency	840
Prerequisites	841
Procedure	841
How Boundary Routing System Architecture Works	844
Where Can Boundary Routing Be Used?	844
Typical Boundary Routing Environment	846
Non-IBM Environment Using a NETBuilder II Bridge/Router	847
Non-IBM Environment Using a SuperStack II Bridge/Router Model 227 or 427	848
IBM Environment Using a NETBuilder II Bridge/Router as a Central Node	850
IBM Environment Using a NETBuilder II Bridge/Router as a Regional Central Node	852
IBM Environment Using a SuperStack II NETBuilder Bridge/Router Model 327 or 527 As a Central Node	854
APPN Topology	855
SDLC Over Boundary Router Links	856
Boundary Routing Features	856
Simplified Network Administration	856
Reduced WAN Usage Costs	856
Increased Reliability	860
Continuous Operation	862
Dual PVCs for IBM Traffic	865
Network Resiliency	865
Network Resiliency Using a Redundant Link	866
Network Resiliency Using a Redundant Route to an Alternate Central Node	871
Using the Central MAC Address	878

CONFIGURING AUTOSTARTUP

Prerequisites and Tools	881
Preparation	882
Tools	882
Prerequisites	882
Configuring the Central Site Network Management Station	883
Procedure	884
Autostarting the Central Site Node	888
How Autostartup Works	888
Autostartup Phase 1	888
Automatic Attribute Detection for DTE Ports on Remote Bridge/Routers	889
Autostartup Phase 2	890
Sample Configurations	890
BootP Server	890
3Com Nonproprietary BootP Servers	890
Sample Configuration: Frame Relay WAN	891
Sample Configuration: PPP WAN	892
Cisco Router at the Central Site	893

CONFIGURING WITH ASCII FILES

Overview of ASCII File Usage	895
ASCII Text Configuration Files	895
Downloading the ASCII Text Configuration File	895
Executing the Configuration File with LoadConfigs	896
Additional Configuration File Uses	896
ASCII Boot	896
AutoStartup	896
ASCII Capture	896
ASCII Boot	896
Creating the ASCII Text File	897
Downloading the ASCII File to the Device	897
Executing boot.cfg	897
The CONFIG.LOG File	898
Renaming the boot.cfg File	898
Limitations of ASCII Boot Feature	898
Basic Configuration Procedure	899
ASCII Capture	902
Limitations	903
SNMP Command	903
Procedure	903
Flushing the Cache	904
Reviewing the Capture File	904
Example 1	904
Example 2	904
How Passwords are Captured	905
Example 3	905

CONFIGURING WIDE AREA NETWORKING USING PPP

Configuring Point-to-Point Protocol Communication	907
Enabling PPP	908
Setting an Authentication Protocol	908
Setting Up PAP	908
On the Local Bridge/Router 1	909
On the Remote Router 2	909
Setting Up CHAP	910
Setting Up MS-CHAP	910
Verifying Your Configuration	911
Setting Up EAP	911
Activating LAPB to Reduce Noisy Lines	912
How PPP Works	913
Packet Size Negotiation	913
Serial Line Management	914
Serial Line Quality Maintenance	914
How Authentication Works	914
Load Sharing and Load Balancing	915

CONFIGURING WIDE AREA NETWORKING USING ISDN

Planning Your ISDN Network	918
Deciding How to Use the ISDN Interface	919
Disabling Phantom Power	922
Setting Up the Remote Device	922
How the ISDN Interface Works	922
Basic Rate Interface	922
Point-to-Point and Point-to-Multipoint Configurations	923
How Incoming Calls Are Accepted	923
Bearer Capability Compatibility	924
ISDN Addressing Compatibility	924
ISDN Addressing	927
Austel Semi Permanent Circuit Support	928

CONFIGURING THE NETBUILDER II TO USE A WAN EXTENDER

Circuit Services Supported	931
Configuring WAN Extender and NETBuilder II for Remote Connections Requirements	931
Interconnecting Leased DS0s to Channelized T1	932
Configuring the WAN Extender	933
Configuring the NETBuilder II Bridge/Router	934
Configuring Other Protocols	936
Verifying the Configuration	936
Interconnecting ISDN BRI Circuits to ISDN PRI	937
Configuring the WAN Extender	938
Configuring the NETBuilder II Bridge/Router	939
Configuring Other Protocols	942
Verifying the Configuration	942
Configuring Switched 56 Circuits	942
Remote Connection Configuration Considerations	943
Dial-Up Options	943
Operator-Initiated Dialing (Manual Dial)	943
Scheduled Dial	943
Auto Dial	943
Dial-on-Demand	943
Remote Site Identification Options	944
ISDN Caller ID on the WAN Extender	944
ISDN Called ID on the WAN Extender	944
PPP System ID Data on the NETBuilder II Bridge/Router	944
CLIP Service Configuration	945
Customizing the Configurations	945
ISDN H0 Support (WAN Extender 2T Only)	945
Call Filtering	945
Channel Bundling	946
NETBuilder II Configuration Commands and Parameters	946
Commands	946
DLTest	946
PATH Service Parameters	946
Baud	946
CLock	946
CONFiguration	947
CONNector	947
CONTRol	947
DialCONTRol	947
DialPool	948
ExDevType	948
LineType	948
PORT Service Parameters	948
CLList	948
COMPRESSType	949
CONFiguration	949

DialNoList	949
DialRcvrState	950
DialStatus	950
OWNer	950
PAtHs	950
PathPreference	950
VirtualPort	950
Sample Configuration Verification Displays	950
Configuration Setting Displays	951
Connection and Data Packet Statistics Displays	951
Incoming and Outgoing Calls Displays	952
Packet Counts Displays	952
Troubleshooting	953
Troubleshooting Channelized Leased Configurations	953
Troubleshooting Switch Circuit Configurations	953
Using WAN Extender Troubleshooting Commands	954
Accessing the WAN Extender Console Interface	954
Command Descriptions	954
Using NETBuilder II Troubleshooting Commands	957
WAN Extender Service Parameters	957
How the WAN Extender Works	960
WAN Extender Models	960
How Virtual Paths are Created	960
Leased Virtual Paths	961
DS0 Dial Virtual Paths	961
H0 Virtual Paths	961
How the WAN Extender Operates	962

CONFIGURING PORT BANDWIDTH MANAGEMENT

Communication Resources Supported	965
Sync Dial Lines	966
Integrated and Internal ISDN Lines	966
WAN Extender Virtual Paths	966
Associating Paths to Ports	966
Static versus Dynamic Paths	966
Dynamic Dial Path Pooling	967
Valid Port and Path Configurations	968
System Bandwidth Management	968
Dial-on-Demand	968
Bandwidth-on-Demand	969
Disaster Recovery	969
Path Configuration Summary	970
Resource Aggregation	970
Dial Number List	970
Prioritized Path Preferences	970
Manual Bandwidth Management	971
Manual Dial	971

Manual Hangup	972
Manual Bandwidth Management Disaster Recovery	972
Bandwidth Management Status Displays	972
Bandwidth Management Statistical Displays	972
Configuring Wide Area Networking Using Async PPP	972
Configuring Async PPP and AT Dial	972
Prerequisites	972
Procedure	973
Initializing the Modem	973
Adding a Number to the Phone Number List	974
Modem Initialization Strings	975
Command Interaction	976
Signal Interaction	976
Online Operation	977
Example Initialization Strings	978
Configuring WAN Resources	978
Configuring Dial-Up Lines Using a Modem or TA	978
Prerequisites	978
Procedure	978
Configuring ISDN Lines	980
Prerequisites	980
Procedure	981
Configuring Leased Lines	982
Prerequisites	982
Procedure	982
Configuring ASPC Leased Lines	983
Prerequisites	983
Procedure	983
Configuring System Bandwidth Management Mode (DOD)	984
Prerequisites	984
Procedure	984
Configuring Bandwidth-on-Demand	984
Prerequisites	984
Procedure	984
Configuring the Dial List	985
Prerequisites	985
Procedure	985
Adding a Phone Number	986
Editing an Existing Phone Number	987
Deleting a Phone Number	987
Binding Paths to Ports	987
Converting a Static Path to a Dynamic Path	987
Changing a Dynamic Path to a Static Path	987
Identifying Remote Sites with SCID and CLIP	988
Configuring a Port to Use SCID	988
To Configure a Port to Use CLIP	989
Configuring PAP, CHAP and Standard Bundling	989

Configuring Point to Point Tunneling Protocol	989
Configuring the Path Preference List	989
Prerequisites	989
Procedure	990
Appending a Path	991
Adding a Path	991
Deleting a Path	992
Configuring Manual Bandwidth Management Mode	992
Prerequisites	992
Procedure	992
Disaster Recovery Procedure	993
Verifying the Configuration	993
Troubleshooting the Configuration	994
Configuration Examples	994
Load Balancing over Multiple Dial-up Links	994
NETBuilder II WAN Extender Configuration Example	995
Routing Configurations over DOD Links	997
IP over a DOD Link	997
RIP over a DOD Link	998
TCP for SNA Traffic over a DOD Link	998
IPX with Incremental Broadcasts over a DOD Link	999
IPX Protocol in a Boundary Routing Environment over a DOD Link	1000
Example 1	1000
Example 2	1000
Summary of Bandwidth Manager Commands and Parameters	1001
Bandwidth Management Concepts	1004
Virtual Pipe	1004
Bandwidth	1004
Bandwidth Aggregation	1005
Bandwidth Management Terms	1005

CONFIGURING WIDE AREA NETWORKING USING FRAME RELAY

Setting Up the Frame Relay Service	1007
Prerequisites	1007
Procedure	1008
Configuring Congestion Control	1008
For NTTLMI Protocol Users	1008
For Other LMI Protocol Users	1009
Configuring PVCs and SVCs	1010
Verifying the Configuration	1012
Setting Up Basic Bridging over Frame Relay	1012
Configuring Transparent Bridging	1012
Prerequisites	1012
Procedure	1012
Configuring Source Route Bridging	1013
Prerequisites	1013

Procedure	1013
Setting Up Basic Routing over Frame Relay	1013
Configuring AppleTalk	1014
Prerequisites	1014
Non-AppleTalk Configuration	1015
AppleTalk Configuration	1015
Configuring APPN	1016
Prerequisites	1016
Procedure	1016
Configuring APPN with Virtual Ports	1018
Deleting APPN Virtual Ports	1019
Configuring DECnet	1019
Prerequisites	1019
Procedure	1020
Configuring IP	1020
Prerequisites	1021
Procedure	1021
Configuring IPX	1023
Prerequisites	1023
Procedure	1023
Configuring OSI	1025
Prerequisites	1025
Procedure	1026
Configuring VINES	1026
Prerequisites	1026
Procedure	1027
Configuring XNS	1027
Prerequisites	1027
Procedure	1028
Configuring Disaster Recovery	1028
Prerequisites	1029
Procedure	1029
Configuring a Primary PVC	1030
Configuring a Backup PVC	1030
Configuring a Backup Link	1031
How Frame Relay Works	1031
PVC and SVC Connections	1032
Establishing a PVC Connection	1032
Establishing an SVC Connection	1033
Fully Meshed, Partially Meshed, and Nonmeshed Topologies	1034
Frame Relay Addresses	1037
Local Management Interface Protocol	1038
How Disaster Recovery Works	1039
Using Virtual Ports for Disaster Recovery	1039
Partially Redundant Networks	1040
Fully Redundant Networks	1041
Frame Relay Congestion Control	1042
How Congestion Control Works	1042

Frame Relay Congestion Control, LLC2, and SNA	1045
Frame Relay Auto Startup	1045

CONFIGURING WIDE AREA NETWORKING USING SMDS

Setting Up the SMDS Service	1047
Prerequisites	1048
Procedure	1048
Verifying the Configuration	1049
Setting Up Basic Bridging over SMDS	1049
Configuring Transparent Bridging	1049
Prerequisites	1049
Procedure	1050
Configuring Source Route Bridging	1051
Prerequisites	1051
Procedure	1051
Setting Up Basic Routing over SMDS	1052
Configuring AppleTalk	1052
Prerequisites	1052
Procedures	1053
Group Address Configuration	1054
Individual Address Configuration	1054
Configuring DECnet	1056
Prerequisites	1056
Procedure	1056
Configuring IP	1057
Prerequisites	1057
Procedure	1058
Configuring IPX	1060
Prerequisites	1060
Procedure	1060
Configuring OSI	1062
Prerequisites	1062
Procedure	1062
Configuring VINES	1064
Prerequisites	1064
Procedure	1064
Configuring XNS	1065
Prerequisites	1065
Procedure	1065
How SMDS Works	1066
SMDS Addresses	1067
Local Management Interface Protocol	1067
SMDS Service Limits	1067
Separating Routing Protocols	1068
Transparent Bridging	1068
Source Route and Transparent Bridge Separation	1071
AppleTalk Route Filtering	1071

IPX Migration from RIP/SAP to NLSP	1071
IP Route Policy	1071
Large Hierarchical Networks	1072

CONFIGURING WIDE AREA NETWORKING USING X.25

Setting Up the X25 Service	1073
Prerequisites	1074
Procedure	1074
Verifying the Configuration	1075
Using X.25 Profiles	1075
User Profiles	1075
DTE Profiles	1076
X.25 Profile Parameter Usage	1076
Configuration Parameters	1077
X.25 Profiles Configuration Examples	1078
1080	
Setting Up Basic Routing over X.25	1081
Configuring AppleTalk	1082
Non-AppleTalk Prerequisites	1082
Non-AppleTalk Procedure	1082
AppleTalk Prerequisites	1083
AppleTalk Procedure	1084
Configuring DECnet	1085
Prerequisites	1085
Procedure	1085
Configuring IP	1086
Prerequisites	1086
Procedure	1087
Configuring IPX	1089
Prerequisites	1089
Procedure	1090
Configuring IPX with Different Software Versions	1092
Configuring OSI	1092
Prerequisites	1093
Procedure	1093
Configuring VINES	1094
Prerequisites	1094
Procedure	1095
Configuring XNS	1095
Prerequisites	1096
Procedure	1096
Procedure	1097
Setting Up Bridging over X.25	1098
Configuring Transparent Bridging	1098
Prerequisites	1098
Procedure	1099
Configuring Source Route Bridging	1100

Prerequisites	1100
Procedure	1100
Setting Up a Permanent Virtual Circuit Connection	1102
Prerequisites	1102
Procedure	1102
How X.25 Works	1102
Fully Meshed, Partially Meshed, and Nonmeshed Topologies	1102
Facilities	1105

CONFIGURING LOCAL AND GLOBAL SWITCHING

Setting Up Local Switching on a SVC	1107
Setting Up Global Switching on an SVC	1108
Setting up Local Switching on a PVC	1109
Setting up Global Switching on a PVC	1110
Configuring the Local-end Router	1111
Configuring the Remote-end Router	1112
Setting up Switching on a PVC Over a WAN	1114
Configuring Local Router A	1115
Configuring the Remote Routers	1115
Configuring Remote Router B	1115
Configuring Remote Router C	1115
Configuring Remote Router D	1116
Switching Terms	1116

CONFIGURING CONNECTIONS FOR OUTGOING CALLS

Setting Up the Gateway for Outgoing Telnet Connections	1117
Prerequisites	1117
Procedure	1117
Setting Up the Gateway for Outgoing VTP Connections	1122
Prerequisites	1122
Procedure	1122
Making Outgoing Connections	1125
Automatic Connections	1126
Extended Connections	1126
Selecting Individual PAD Parameters	1127
Requesting Current Values of PAD Parameters	1127
Establishing a Virtual Call	1127
Clearing a Virtual Call	1129
Troubleshooting Outgoing Connections	1129
How the Outgoing Connection Service Works	1130

CONFIGURING CONNECTIONS FOR INCOMING CALLS

- Configuring the Gateway for Incoming Connections 1133
 - Prerequisites 1133
 - Procedure 1134
- Making Incoming Connections 1134
 - Automatic Connections 1134
 - Using Addresses 1135
 - Using Names 1135
 - Using Configuration Files 1135
 - Extended Connections 1136
- Troubleshooting Incoming Connections 1137
- Customizing the Incoming Connection Service 1138
 - Creating Port-Initialization Macros 1138
 - Creating Macros 1139
 - Assigning the Macro to a Configuration File 1140
 - Managing Macros 1141
 - Name Service for TCP/IP Connections 1141
 - Domain Name Service 1142
 - Configuring Rlogin Connections 1143
 - Name Service for OSI Connections 1143
 - X.500 Directory Service 1145
 - File-Based Name Service 1152
- How the Incoming Connection Service Works 1153

CONFIGURING LOCAL ACCESS CONTROL

- Configuring Local Access Control 1155
 - Procedure 1155
 - Related Information 1156
 - Logging On and Logging Out (in the CX package) 1156
 - Changing User Passwords 1156

MANAGING SESSIONS FOR INCOMING EXTENDED CALLS

- Making Connections to IP Internet-attached and OSI Hosts 1159
 - Making Connections with the Connect Command 1159
 - Making Telnet Connections to TCP/IP Resources 1161
 - Making Rlogin Connections to Resources 1163
 - Making Connections to OSI Resources 1164
- Troubleshooting Connection Error Messages 1166
- Checking Network Resources 1167
 - Checking TCP/IP Network Resources 1167
 - Checking OSI Network Resources 1168
- Managing Sessions 1169
 - Establishing a Single Session 1169
 - Establishing Multiple Sessions 1170
 - Displaying Session Information 1171
 - Changing the Current Session 1171
 - Moving between Sessions 1171
 - Using the RESume Command 1172
 - Using the FORwards and BACKwards Commands 1172
 - Using the ECM Character to Enter Command Mode 1172
 - Disconnecting a Single Session 1173
 - Disconnecting Multiple Sessions 1173
 - Changing Session Parameters 1173

CONFIGURING INTERNETWORKING USING ATM

- Setting Up the ATM Service 1175
 - Prerequisites 1175
 - Procedure 1176
- Verifying the Configuration 1178
- Monitoring the Network 1178
- Configuring Transparent Bridging 1179
 - Prerequisites 1179
 - Procedure 1179
- Configuring Source Route Bridging 1180
 - Prerequisites 1180
 - Procedure 1180
- Configuring IP Routing 1181
 - Prerequisites 1181
 - Procedure 1182
- Configuring IPX Routing 1184
 - Prerequisites 1184
 - Procedure 1184
- How ATM Works 1186
 - Network Interfaces 1186
 - ATM Addressing, Virtual Paths, and Virtual Channels 1187
 - Encapsulation Types 1188
 - Quality of Service 1188

Traffic Shapers	1189
Outbound Data Traffic Control	1190
Bandwidth Reservation	1190
Prioritization of Traffic among VCCs of the Same Protocol	1191
Prioritization of Traffic among VCCs of Different Protocols	1191
Network Management	1191
Fully Meshed, Partially Meshed, and Nonmeshed Topologies	1192
ATM Terms	1195

CONFIGURING INTERNETWORKING USING ATM AND LAN EMULATION

Setting Up the ATMLE Service	1197
Prerequisites	1197
Procedure for Ethernet LANE	1197
Procedure for Token Ring LANE	1198
Setting Up LAN Emulation Client Source Routing	1198
Verifying the Configuration	1200
Controlling Initialization	1200
Configuring Multiprotocol Over ATM Services	1200
Procedure	1201
How ATM and LAN Emulation Work	1203
Network Interfaces	1204
ATM Addressing	1204
LAN Emulation	1204
LUNI Components and Connections	1204
LAN Emulation Client	1205
LAN Emulation Configuration Server	1205
LAN Emulation Server	1205
Broadcast and Unknown Server	1205
Operation	1205
Initialization and Configuration	1205
Joining and Registration	1206
Data Transfer	1206
Multiprotocol Over ATM Background	1207
Token Ring LAN Emulation Client	1208
Source Routing	1209
RD Registration	1209
Data Transfer	1209
ATM LAN Emulation Terms	1210

CONFIGURING WIDE AREA NETWORKING USING THE ATM DXI

Configuring ATM DXI	1214
ATM Address Mapping	1214
Encapsulation Type and AAL Support	1215
LMI Protocol	1215
Setting Up the ATM Service	1215
Configuring Transparent Bridging	1215
Configuring IPX over an ATM Network	1215
Configuring XNS over an ATM Network	1215
How ATM DXI Works	1216
Address Mapping	1216
Encapsulation Type	1216

CONFIGURING FDDI

Configuring Ports for FDDI	1217
Troubleshooting the Configuration	1217
Diagnosing Internal Hardware Problems	1217
Diagnosing Network Problems	1218

CONFIGURING MNEMONIC FILTERING

Configuring Filters	1221
Using Built-in Masks	1222
Using User-defined Masks	1222
Grouping Related Stations	1223
Parameter Overview	1225
How Filtering Works	1225
Selection	1226
Qualification	1227
Action	1227
Count	1227
Discard	1227
DodDiscard	1227
Forward	1228
PROTocolRsrv <tag>	1228
Sequence	1228
Prioritization (Priority Queuing)	1229
Trace	1229
Built-in Bridge Masks	1229
Built-in IPX Masks	1230
Built-in IBM Trace Masks	1230
User-defined Bridge Masks	1230
User-defined IPX Masks	1232
Bridge Filtering Examples	1233
IPX Filtering Examples	1241
Setting Up IPX Filter Masks	1241

CONFIGURING PROTOCOL RESERVATION

Why Use Protocol Reservation	1247
Protocol Reservation Procedural Overview	1249
Using Protocol Reservation with Frame Relay Virtual Ports	1251
Configuring for Bridged Traffic or IP- or IPX-Routed Traffic	1252
Configuring for Bridged Traffic	1252
Configuring for IP-Routed Packets	1253
Prerequisites	1253
Procedure	1254
How Protocol Reservation Allocates Different IP Protocol Types	1255
Configuring for IPX-Routed Traffic	1256
Configuring for IBM Traffic	1257
Configuring for DLSw Traffic at the Tunnel Endpoint	1258
Configuring for LLC2 Traffic for SNA Boundary Routing	1259
Configuring for APPN-Routed Traffic	1261
Protocol Reservation Configuration Examples	1262
Example 1: Mixed Bridged Traffic	1262
Example 2: Mixed-Routed Packets	1264
Example 3: Virtual Ports	1265
How Protocol Reservation Works	1265
How Protocol Reservation Controls Bandwidth for Traffic Types	1266
Tuning	1266
Bandwidth Allocation Process Rules	1266
Bandwidth Normalization	1266
Distribution of Non-Allocated Bandwidth	1267

CONFIGURING DATA COMPRESSION

Configuring Data Compression	1269
Configuring Tinygram Compression	1269
Configuring Link-Level Compression	1269
Enabling History-based or Per-packet Compression	1270
Optional Configurations	1270
Enabling LAPB for a PPP Link	1270
Frame Relay Configuration Options	1270
X.25 Configuration Options	1271
Verifying Link-Level Compression Effectiveness	1271
How Data Compression Works	1272
Tinygram Compression	1272
Link-Level Compression	1272
When To Use Tinygram Compression	1273
When To Use Link-Level Compression	1273

PRIORITIZING MULTIPROTOCOL DATA

- Advantages of Prioritizing Data 1275
- Setting Up Data Prioritization 1275
 - Prerequisites 1275
 - Procedure 1276
- Prioritizing LLC2-, SNA-, and NetBIOS-Bridged Packets 1277
 - Prioritizing LLC2-Bridged Packets From Two Groups of End Stations 1278
 - Prioritizing SNA- and NetBIOS-Bridged Packets **1278**
 - Assigning a Priority to Different IP Packets 1279
- Data Prioritization Parameters 1279
- How Data Prioritization Works 1280
 - How Packets Are Assigned a Priority 1281
 - Queues 1281
 - Queue Arbitration Algorithm 1283

NETWORK MANAGEMENT

- Simple Network Management Protocol 1285
 - Configuring the SNMP Service 1285
 - Procedure 1286
 - Related Information 1286
 - Request Validation 1286
- Remote Network Monitoring Alarms 1287
- Network Maps 1288
- Logging Configuration Changes 1289
 - Configuring Multiple Syslog Servers 1291
 - Managing AuditLog Filters 1292
 - Defining a Filter Using the ADD LogFilter Command 1292
 - Displaying Filters 1292
 - Deleting Filters 1292
 - AuditLog Filter Examples 1292
 - Log All Events of Severity Level 5 (Notification) and Above to the Default Server 1293
 - Suppress All ISDN UP/DOWN Events When Severity Level 3 (Error) and Above Are Sent to the Default Server 1293
 - Send All VPN-related Syslog Messages to Server 100.100.1.2 and All Other Syslog Messages to All Servers 1293
 - Suppress Sending All Syslog Messages Resulting from UI Commands to the Default Server 1293
 - Send Events with Severity Level 4 to Server 100.100.1.1 and Events with Severity Level 5 to Server 100.100.1.2 1294
- SNMP Event Notification Traps 1295
- Remote Access of Your System 1295
 - Using the REMote Command or the TELnet Command 1296
 - Preventing Remote Access 1297
 - Restricting Remote Access 1298
 - Restricting Telnet Access 1298
- Resynchronization Feature for Encryption Devices 1298

LAN Net Manager Support	1299
Configuring LAN Net Manager Support	1299
Configuring Virtual Bridges and a Virtual Ring for NETBuilder II	1300
Disabling LAN Net Manager Support	1301
AMP-Based Network Device Discovery	1301
Configuring the Discovery Responder	1302
Configuring AMP Using the BRidge Service	1302

SCHEDULING AND EVENT-BASED MACRO EXECUTION

Creating Schedules	1305
Defining a Daily Schedule	1305
Creating an Active Schedule	1305
Executing Macros Using the Scheduler	1305
Scheduling WAN Connections	1306
Executing Event-based Commands/Macros	1306
Setting Up a Backup Port	1307
Hanging Up a Port	1307
Recovering from Port Loopback	1307
How the Scheduler Works	1308
How EBME Works	1308

SWAPPING NETBUILDER II HARDWARE MODULES

Swapping Hardware Modules	1311
---------------------------	------

DIAL-UP PROGRESS AND ERROR MESSAGES

HSS Line Driver Cards	1313
DTE Connector Transmit and Receive States	1313
Dial-Up Progress and Error Messages	1313
Software Messages for Modems	1313
V.25 Modems	1314
Software Messages for SuperStack II NETBuilder Bridge/Router	1314

LOOPBACK TESTING

Dial-up Loopback Testing Using Modems	1319
Performing a Local Loopback Test	1321
Performing a Remote Loopback Test	1322
Making the Loopback Fixture	1323
Loopback Testing for Built-In ISDN Ports	1324
Procedure	1324

INTERNET ADDRESSING

Internet Addresses	1327
Class A Address Format	1327
Class B Address Format	1328
Class C Address Format	1328
Class D Address Format	1328
Dotted Decimal Notation	1329
Addressing Rules	1329
Sample Network Using the Class B Address Format	1330
Subnet Addresses and Subnet Masks	1330
Subnet Addressing	1330
Regular Internet Address Format	1331
Subnet Address Format	1331
Subnet Masks	1332
Subnet Address Format	1332
Subnet Mask	1332
Subnet Address Format for 128.121.61.100	1332
Subnet Mask	1332
Subnets: Example 1	1333
Subnets: Example 2	1335
Subnets: Example 3	1336
Variable Length Subnet Masks	1338

NSAP AND PSAP ADDRESSING

NSAP Address Structure	1339
NSAP Address Assignment	1340
Default NSAP Values	1341
Values Derived from NSAP Addresses	1341
NSAP Registration Authorities	1341
PSAP Addresses	1342
NSAP and PSAP Address Field Definitions	1343

SUPPORTED MIBs

Supported Operations	1345
Port Numbering Convention in SNMP	1345
MIBs Supported by the Bridge/Router	1346
3Com Private MIBs	1348

MACRO FEATURES

- Macro Conventions 1349
- Macros With Conditional Statements 1349
 - Macro Variables 1349
 - Variable Types 1350
 - Comparing and Reassigning Variables 1353
 - Variable Substitutions 1354
 - Control Structures 1354
 - If-Else-End 1354
 - Switch-Case-End 1355
 - Loop-End 1355
 - Keywords 1355
 - Audit 1356
 - Break 1356
 - Continue 1356
 - Exit 1356
 - Return 1356
 - Macro Caching and Shared Macros 1356
 - Larger Macros 1357
 - Macro Nesting 1358

STATISTICS DISPLAYS

- AppleTalk Service 1359
 - DDP Statistics 1360
 - General Datagram Counts 1360
 - Dropped Datagram Counts 1361
 - RTMP Statistics 1361
 - ZIP Statistics 1362
 - AEP Statistics 1363
 - NBP Statistics 1363
- ARP Service 1363
 - Data Pkts Discarded 1364
 - Data Pkts In Queue 1364
 - Requests Received 1364
 - Requests Sent 1364
 - InARP Statistics 1365
 - RARP Statistics 1365
- ATUN Service 1365
- BGP Service 1366
 - BGP Statistics for All Peers 1366
 - Per-peer Statistics 1366
- BRidge Service 1367
- BSC Service 1368
- CLNP Service 1369
 - CLNP statistics 1369
 - Rcvd: good PDU 1369

- Xmit: good PDU 1369
- DECnet Service 1370
 - Data Messages 1370
 - Routing Messages 1370
 - Hello Messages 1371
 - Phase V Data Messages 1371
 - Internetwork Data Messages 1371
- DLSw Service 1371
- DVMRP Service 1373
 - DVMRP Statistics 1373
 - Pkts Received 1373
 - Pkts Transmitted 1373
 - Pkts Forwarded 1374
 - Pkts Discarded 1374
 - IP over IP Statistics 1374
- FR Service 1374
 - Frame Relay Port Statistics 1374
- IDP Service 1374
 - IDP Statistics 1374
- IP Service 1375
 - IP Statistics Descriptions 1377
 - IPX Statistics 1381
 - 1381
 - IPX SPOOF Statistics 1382
- ISIS Service 1382
- LLC2 Service 1384
 - Test Frames 1384
 - Xid Frames 1384
 - UI-Data Frames 1385
 - Sabme Frames 1385
 - I-Data Frames 1385
 - I-Data Bytes 1385
 - RR Frames 1385
 - RNR Frames 1385
 - Reject Frames 1385
 - Disc Frames 1385
 - UA Frames 1385
 - DM Frames 1385
 - FRMR Frames 1385
- MIP Service 1386
 - Multicast IP Datagram 1386
 - Pkts Received 1386
 - Pkts Transmitted 1386
 - Pkts Discarded 1386
- MOSPF Service 1386
 - MOSFP Statistics 1387
 - Receive 1387
 - Transmit 1387

- NLSP Service 1387
 - NLSP statistics 1388
 - Authentication 1388
- NRIP Service 1389
 - NRIP statistics 1389
- OSPF Service 1390
 - OSPF Statistics 1390
 - Errors 1391
- PATH Service 1391
 - Rcvd Packets 1392
 - Xmit Packets 1392
 - Discard 1393
 - Xmit Good 1393
- PORT Service 1393
 - Rcvd 1393
 - Xmit 1394
 - Filter 1394
 - Discard 1394
 - DialOnDemand Mode 1394
- PPP Service 1394
 - LCP path statistics 1395
 - Rcvd 1395
 - Xmit 1395
 - PPP Over Ethernet Statistics 1396
- RIPIP Service 1396
 - Incoming Packets 1396
 - Outgoing Pkts 1396
- RIPXNS Service 1397
- RSVP Service 1397
 - RSVP Port Statistics 1398
- SAP Service 1398
 - SAP Statistics 1399
- SHDlc Service 1400
 - Frames 1400
 - Bytes 1400
 - Frames Discarded 1400
 - Circuit Count 1400
- SMDS Service 1400
 - Packets Received 1401
 - Packets Transmitted 1401
 - Error Packets Received 1401
- SNMP Service 1401
 - Incoming SNMP PDUs 1401
 - Outgoing SNMP PDUs 1402
- SR Service 1402
 - RECEIVED 1402
 - TRANSMITTED 1402
 - ERRORS 1403

STP Service	1403
STP statistics	1403
SYS Service	1404
Port	1404
Source	1404
Protocol	1404
TCP Service	1404
TCP Packets	1405
TCP Connections	1405
UDP Service	1405
UDP Statistics	1405
UDPHelp Service	1406
BOOTP/UDP/IP Broadcast Helper Statistics	1406
VIP Service	1406
VINES IP Statistics	1407
VINES ARP Statistics	1407
VINES ICP Statistics	1407
VINES RTP Statistics	1408
WE Service	1408
WE Statistics	1408
Received Frame Errors	1409
X25 Service	1409

STATIC TABLES

AUDIT TRAIL MESSAGES

SYSLOG MESSAGES

REGULAR EXPRESSIONS

AS Filter Examples	1433
GREP Command Examples	1433

X.3 PARAMETERS AND PAD PROFILES

X.3-to-TERM Service Parameter Equivalence	1435
CCITT Simple Standard PAD Profile	1436

WIDE AREA NETWORK SETUP INFORMATION

NETBuilder II I/O Module Placement	1437
T3 Plus Interoperability	1437
HSS Port Utilization Percentage	1437
Serial Line Connectivity	1437
External Device Connections	1437
External Device Cable Length	1438
Serial Line Clocking	1438
Synchronizing the Network Clock	1438
Serial Line Supported Data Rates	1438

APPN CONFIGURATION EXAMPLES

AS/400 Configuration	1439
Example 1: Token Ring Over Physical Ports	1439
Example 2: Frame Relay over Physical Ports	1441
Example 3: Frame Relay over Virtual Ports	1442
IBM PC Support/400 Example	1443
Example 4: Setting Up Connections with a DOS PC	1444
Configuration for DLUs/DLUr	1444
APPN Sense Codes	1445

IBM TRACE FACILITY

Tracing IBM Data Traffic	1455
Tracing DLSw Packets	1455
Displaying DLSw Trace Data	1457
DLSw Filter Examples	1457
Tracing DLSw Packets from a Local MAC Address	1457
Tracing DLSw Packets from a Local SAP	1457
Tracing DLSw Packets from a Remote MAC Address	1457
Tracing DLSw Packets from a Remote SAP	1457
Tracing DLSw Packets from an IP Address	1457
Tracing DLSw Control Message Packets from a Local MAC Address	1457
Tracing DLSw Control Message Packets from a Local SAP	1457
Tracing DLSw Control Message Packets from a Remote MAC Address	1458
Tracing DLSw Control Message Packets from a Remote SAP	1458
Tracing DLSw Control Message Packets from an IP Address	1458
Tracing DLSw Information Message Packets from a Local MAC Address	1458
Tracing DLSw Information Message Packets from a Local SAP	1458
Tracing DLSw Information Message Packets from a Remote MAC Address	1458
Tracing DLSw Information Message Packets from a Remote SAP	1458
Tracing DLSw Information Message Packets from an IP Address	1458
Tracing LLC2 Frames	1459
Displaying LLC2 Trace Data	1460
LLC2 Filter Examples	1460

Tracing LLC2 Packets from a Local MAC Address	1460
Tracing LLC2 Packets from a Local SAP	1460
Tracing LLC2 Packets from a Remote MAC Address	1460
Tracing LLC2 Packets from a Remote SAP	1460
Tracing LLC2 Information Frames from a Local MAC Address	1461
Tracing LLC2 Information Frames from a Local SAP	1461
Tracing LLC2 Information Frames from a Remote MAC Address	1461
Tracing LLC2 Information Frames from a Remote SAP	1461
Tracing LLC2 Unnumbered Frames from a Local MAC Address	1461
Tracing LLC2 Unnumbered Frames from a Local SAP	1461
Tracing LLC2 Unnumbered Frames from a Remote MAC Address	1462
Tracing LLC2 Unnumbered Frames from a Remote SAP	1462
Tracing SDLC Frames	1462
Displaying SDLC Trace Data	1463
SDLC Filter Examples	1463
Tracing SDLC Packets from a Poll Address	1463
Tracing SDLC Information Frames	1463
Tracing SDLC Unnumbered Frames	1463
Tracing SDLC Unnumbered Frames	1464

DLSw, APPN, AND BSC HOST CONFIGURATION EXAMPLES

DLSw Host Examples	1465
Example 1: Configuring a 3745 Host with Dual TIC to Support BAN	1465
Example 2: Configuring a Host to Support Boundary Access Node (BAN) Frame Relay Between a Host and a NETBuilder Bridge/Router	1467
Example 3: Configuring a Host to Support Boundary Network Node (BNN) Frame Relay Between a Host and a NETBuilder Bridge/Router	1468
APPN Host Configurations	1469
Example 4: Defining an Adjacent Link Station for a TIC to a Host	1469
Example 5: Defining a Host as an SDLC Link Station	1472
Example 6: Mapping an SDLC DLUR Link Station to a Host SDLC PU Definition	1474
Example 7: Mapping a Default DLUs to the VTAM Start Options	1475
Example 8: Defining an LU Directory Entry	1476
Example 9: Mapping an SNA Class of Service (COS) to a Specific Transmission Priority	1477
Example 10: Mapping an SNA Class of Service to the APPN Service	1478
BSC Host Example	1480
NetView Run Commands Support	1483

ABBREVIATIONS AND ACRONYMS

TECHNICAL SUPPORT

Online Technical Services	1495
World Wide Web Site	1495
3Com Knowledgebase Web Services	1495
3Com FTP Site	1495
3Com Bulletin Board Service	1496
Access by Analog Modem	1496
Access by Digital Modem	1496
3Com Facts Automated Fax Service	1496
Support from Your Network Supplier	1496
Support from 3Com	1497
Returning Products for Repair	1498

INDEX

3COM CORPORATION LIMITED WARRANTY

ABOUT THIS GUIDE

This guide provides information you need to use Enterprise OS software to operate and configure your device. This guide includes procedures for configuring your software for bridging, routing, and wide area protocols, according to your network needs.

Supported devices include:

- NETBuilder II®
- SuperStack® II NETBuilder
- SuperStack II NETBuilder Boundary Router
- OfficeConnect® NETBuilder
- PathBuilder S5xx Tunnel Switch
- PathBuilder S400 WAN Convergence Switch



If the information in the release notes shipped with your product differs from the information in this guide, follow the release notes.

Before you use the information in this guide, you must first install the device according to the hardware installation instructions. You must then install and configure Enterprise OS software on the device. If you are upgrading, see *Upgrading Enterprise OS Software*. For a new installation, see the appropriate guide for your platform listed in Table 1.

Table 1 Software Installation Guides

Platform	Guide
NETBuilder II	<i>New Installation for Enterprise OS Software</i>
SuperStack II NETBuilder	<i>Using the SuperStack II NETBuilder SI Bridge/Router</i>
SuperStack II NETBuilder Boundary Router	<i>Using SuperStack II NETBuilder SI Boundary Router Software</i>
OfficeConnect NETBuilder	<i>Using the OfficeConnect NETBuilder Bridge/Router</i>
PathBuilder S5xx	<i>Using the PathBuilder S5xx Tunnel Switch</i>
PathBuilder S400	<i>Using the PathBuilder S400 WAN Convergence Switch</i>

For a comprehensive description of Enterprise OS software commands, see *Reference for Enterprise OS Software*.



In this guide, the term bridge/router is used regardless of whether the NETBuilder system is configured as a bridge or a router or both.

Audience Description

This guide is intended for network administrators who:

- Have experience planning, maintaining, and troubleshooting local or wide area networks.
- Are familiar with network protocols, bridging and routing, and network management.
- Are responsible for configuring and operating NETBuilder bridge/routers.

Conventions

Table 2 and Table 3 list conventions that are used throughout this guide.

Table 2 Notice Icons




Icon	Notice Type	Alerts you to...
	Information note	Important features or instructions
	Caution	Risk of personal safety, system damage, or loss of data
	Warning	Risk of severe personal injury

Table 3 Text Conventions

Convention	Description
Syntax	Evaluate the syntax provided and supply the appropriate values. Placeholders for values you must supply appear in angle brackets. Example: <p style="margin-left: 40px;">Enable RIP using:</p> <pre style="margin-left: 40px;">SETDefault !<port> -RIP IP CONTROL = Listen</pre> <p style="margin-left: 40px;">In this example, you must supply a port number for <port>.</p>
Commands	Enter the command exactly as shown in text and press the Return or Enter key. Example: <p style="margin-left: 40px;">To remove the IP address, enter:</p> <pre style="margin-left: 40px;">SETDefault !0 -IP NETaddr = 0.0.0.0</pre> <p style="margin-left: 40px;"><i>This guide always gives the full form of a command in uppercase and lowercase letters. However, you can abbreviate commands by entering only the uppercase letters and the appropriate value. Commands are not case-sensitive.</i></p>
Screen displays	This typeface represents information as it appears on the screen.
The words "enter" and "type"	When you see the word "enter" in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says "type."
(continued)	
[Key] names	Key names appear in text in one of two ways: <ul style="list-style-type: none"> ■ Referred to by their labels, such as "the Return key" or "the Escape key" ■ Written with brackets, such as [Return] or [Esc]. <p>If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example:</p> <p style="margin-left: 40px;">Press [Ctrl]+[Alt]+[Del].</p>

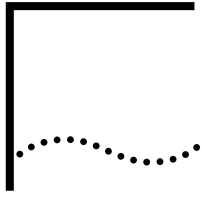
Table 3 Text Conventions (continued)

Convention	Description
<i>Menu commands</i> and <i>buttons</i>	Menu commands or button names appear in italics. Example: From the <i>Help</i> menu, select <i>Contents</i> .
Words in <i>italicized</i> type	Italics emphasize a point or denote new terms at the place where they are defined in the text.
Words in bold-face type	Bold text denotes key features.

Year 2000 Compliance

For information on Year 2000 compliance and 3Com products, visit the 3Com Year 2000 Web page:

<http://www.3com.com/products/yr2000.html>



CONFIGURING BASIC PORTS AND PATHS

To make full use of Enterprise OS software, you need to understand the concept of ports and paths. This chapter provides basic concepts on ports and paths, including numbering, and describes how to configure basic ports and paths on the NETBuilder II® bridge/router, SuperStack® II NETBuilder® bridge/router, and OfficeConnect® NETBuilder bridge/router platforms.



This chapter covers basic port and path concepts only. Some platforms support virtual ports, and the NETBuilder II bridge/router supports port groups (logical networks). If you want to configure virtual ports or port groups, see the Configuring Advanced Ports and Paths chapter.

Concepts

Ports and paths are the fundamental interface units on the bridge/router, and understanding the concept of ports and paths is important. This section defines ports and paths and explains how they are numbered on the following platforms:

- NETBuilder II bridge/router
- SuperStack II NETBuilder bridge/router
- OfficeConnect NETBuilder bridge/router

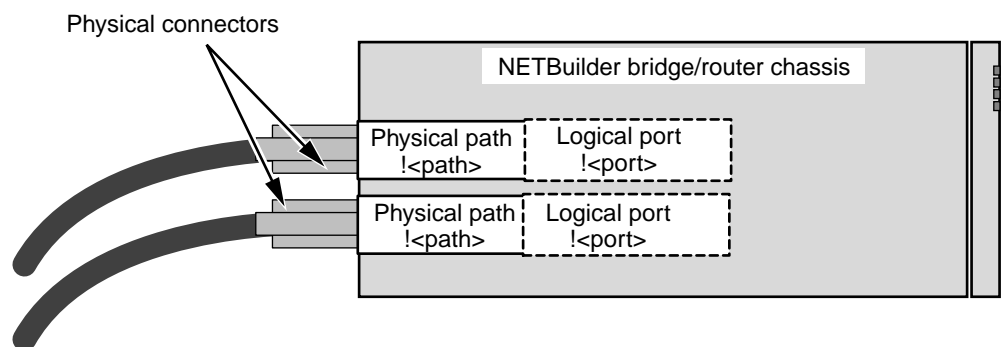
The concepts in this section apply regardless of whether the bridge/router is used as a bridge or as a router.



The local and wide area interfaces available to you depend on your hardware platform and its configuration. For information on the types of interfaces your platform offers, see its installation guide.

The fundamental difference between paths and ports is that the path is the *physical* interface and the port is the *logical* interface in the software that is mapped to the physical path. Figure 1 illustrates the relationship between paths and ports.

Figure 1 Relationship Between Physical Paths and Logical Ports



Paths A *path* is the physical interface that connects a bridge/router to a physical network medium such as an Ethernet bus, a token ring, or a serial line. In an Integrated Services Digital Network (ISDN) environment, a path also represents the channel over which data is transmitted. All NETBuilder bridge/routers provide several paths; each path is associated with a connector, such as an AUI, BNC, RS-232, or RS-449 connector, or a variety of others.

For software purposes, paths are numbered 1, 2, 3, and so on. The path number may be followed by a letter or a decimal and a channel number. For more information, see "Port and Path Numbering on NETBuilder II Multiport Modules" and to "Port and Path Numbering on a SuperStack II Bridge/Router" later in this chapter. For all SuperStack II bridge/router platforms, the connector configuration and the path number for each connector are fixed. For the NETBuilder II bridge/router, a connector takes its path number from the slot in which its module is installed. For more information on NETBuilder II path numbers, see "Port and Path Numbering on a NETBuilder II Bridge/Router" later in this chapter.

Ports A *port* is the logical interface used by the software to represent a connection to a network. By default, there is a one-to-one correspondence between ports and paths, and they are usually numbered alike: for instance, port 1 is associated with path 1. All network traffic received on physical path 1 is treated by the software as arriving on logical port 1, and all traffic that the software transmits through logical port 1 passes through physical path 1. The same is true for the other ports and paths.

This default configuration is called a *static port and path binding*. A *static path* is a path that is mapped to a port. All paths are static by default.

You can redefine the default mapping using software commands. For example, you can redirect network traffic that is being routed through a particular logical port to a different physical path without manually switching cables on the connector.

Each logical port is usually associated with only one physical path. For token ring, Ethernet, Fiber Distributed Data Interface (FDDI), Frame Relay, Asynchronous Transfer Mode (ATM), X.25, and Switched Multimegabit Data Service (SMDS), the path-to-port ratio is always one to one. But for paths connected to serial lines, multiple paths can be associated with and statically bound to a single port if the Point-to-Point Protocol (PPP) is running over the port.

Paths can also be unbound from their ports and placed in a dial path pool to be shared by more than one port. The paths in the dial pool are called *dynamic paths*. A path in the dial pool can be dynamically bound to a port running PPP when the path is needed for data transfer events associated with dial-up. A dynamic path can also be bound to a port for dial backup purposes such as bandwidth-on-demand or disaster recovery. For more information about the use of the dial path pool, see the Configuring Port Bandwidth Management chapter.

Port and Path Numbering on a NETBuilder II Bridge/Router

The configuration of ports and paths on your NETBuilder II bridge/router depends on the combination of I/O modules installed. For information on acceptable I/O module configurations for 4-, 8-, and 8-slot extended chassis NETBuilder II bridge/routers, see the chassis guide that came with your NETBuilder II bridge/router.

Port and Path Numbering on NETBuilder II Multiport Modules

Unlike other I/O modules used on the NETBuilder II bridge/router, the following modules have multiple physical connectors:

- The Ethernet 2-Port 10BASE-FL module has two physical connectors on each board.
- The HSS V.35 3-Port module has an external adapter that maps the single high-density connector on the module into three separate V.35 standard interface connectors.
- The Ethernet 6-Port 10BASE-T module has six connectors on each board.
- The HSS 8-port BRI module has eight connectors on each board.

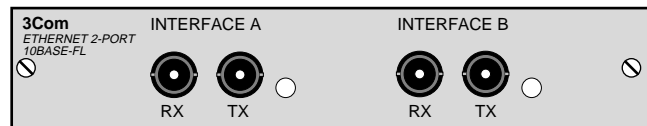


The port and path numbering for the HSS 8-port BRI module is slightly different than for other multiport modules. For more information, see "Port and Path Numbering for the HSS 8-Port BRI Modules" later in this chapter.

Ports and paths on these four multiport modules are labeled differently from ports and paths on single-port modules. To differentiate one physical path or one logical port from another, you append a letter to the path or port number.

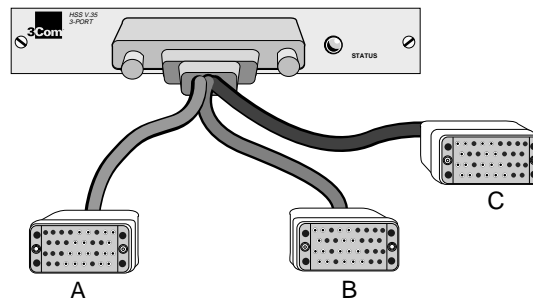
The Ethernet 2-Port 10BASE-FL module has two interfaces labeled A and B (see Figure 2). You add the upper- or lower-case letter A or B to the port number designation. For example, if the module is in slot 4, then the designation for interface B when entering software commands is !4B or !4b.

Figure 2 Ethernet 2-Port 10BASE-FL Module



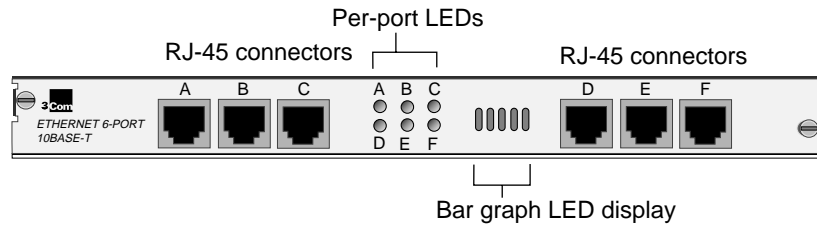
The HSS V.35 3-Port module has three interfaces labeled A, B, and C (see Figure 3). The designations correspond to the markings on the V.35 adapter cable.

Figure 3 HSS V.35 3-Port Module



The Ethernet 6-Port 10BASE-T module has six connectors labeled A through F (see Figure 4).

Figure 4 Ethernet 6-Port 10BASE-T Module



To configure port or path settings on a multiport module, you must identify the slot number and the interface letter. For example, to configure settings for the Ethernet 2-Port 10BASE-FL module in slot 4 of the NETBuilder II hardware, use the path 4B. To assign a name to this path, enter:

```
SETDefault !4B -PATH Name = "SJOSE"
```

To configure settings for interface A, you can use either 4 or 4A. For example, to name path 4A, enter one of the following commands:

```
SETDefault !4 -PATH Name = "SFRAN"
SETDefault !4A -PATH Name = "SFRAN"
```

For more information regarding multiport modules, see the module installation guides.

Port and Path Numbering for the HSS 8-Port BRI Modules Because the HSS 8-Port BRI modules use built-in ISDN interfaces, the port and path numbering convention is different from other multiport modules.

The 8-Port BRI S/T and 8-Port BRI U modules have eight connectors labeled A through H (see Figure 5).

Figure 5 8-port BRI Module



The port and path numbers for the 8-port BRI modules have three components: the slot number, the interface letter (specified with the appropriate letter A through H), and the channel number. For example, the path number for interface A on slot 1 with channel 1 would be 1A.1. To enable the path, enter:

```
SETDefault !1A.1 -PATH CONTROL = Enable
```

Or, the path number for interface H on slot 6 with channel 2 would be 6H.2. To enable this path, enter:

```
SETDefault !6H.2 -PATH CONTROL = Enable
```



The special HSS 8-port BRI module command syntax applies to both the S/T and U BRI modules. The same conventions also apply to port numbers.

By default, 8-port BRI module ports are included in the dial pool when the bridge/router is reset. If you do not want the ports to be in the dial pool, you must change them to the static path and bind the port to the paths using:

```
SETDefault !<connectorID.channelID> -PATH DialCONTROL = STatic
ADD !<port> -PORT PATHs <connector.ID.channelID [,...]> |
    SCID"<SyscallerID>"
SETDefault !<path> -PATH CONTROL = Enable
SETDefault !<port> -PORT CONTROL = Enable
```

For example, to remove port 4A.2 from the dial pool and bind it to a static path, enter:

```
SETDefault !4A.2 -PATH DialCONTROL = STatic
ADD !4A.2 -PORT PATHs !4A.2
SETDefault !4A.2 -PATH CONTROL = Enable
SETDefault !4A.2 -PORT CONTROL = Enable
```

For more information about ISDN port and path numbering, see "Port and Path Numbering Issues for Built-in ISDN Interfaces" later in this chapter. For more information about the HSS 8-port BRI modules, see *Installing the NETBuilder II HSS 8-Port BRI Module*.

Port and Path Numbering on a SuperStack II Bridge/Router

Table 1, Table 2, Table 3 and Table 4 outline the default port and path numbering for model 2xx, 42x, 32x, and 52x SuperStack II NETBuilder bridge/routers.

Table 1 Path and Port Numbering for Model 2xx SuperStack II Bridge/Routers

Physical Path No.	Connector Mapped To	Logical Port No. Mapped To
1	10BASE-T or AUI (Depends on which connector is cabled.)	1
2	V.35	2
3	RS-449	3
4	RS-232	4

Table 2 Path and Port Numbering for Model 42x SuperStack II Bridge/Routers

Physical Path No.	Connector Mapped To*	Logical Port No. Mapped To
1	10BASE-T or AUI (Depends on which connector is cabled.)	1
2.1	ISDN	2
2.2	ISDN	3
3	V.36/RS-449 or RS-232 (Depends on which connector is cabled. Use only one of these connectors at a time.)	4

* The connector associated with paths 2.1, 2.2 cannot be reconfigured.

Table 3 Path and Port Numbering for Model 32x SuperStack II Bridge/Routers

Physical Path No.	Connector Mapped To	Connector Label	Logical Port No. Mapped To
1	UTP or STP (Depends on which connector is cabled.)	UTP or STP	1
2	V.35	A	2
3	Universal serial connector (USC)*	B	3
4	RS-232	C	4

* This connector can be converted to an X.21, V.35, V.36, RS-449, or RS-232 connector using cables. For more information, see your SuperStack II installation guide.

Table 4 Path and Port Numbering for Model 52x SuperStack II Bridge/Routers

Physical Path No.	Connector Mapped To	Connector Marking	Logical Port No. Mapped To*
1	UTP or STP (Depends on which connector is cabled)	UTP or STP	1
2.1	ISDN	ISDN	2
2.2	ISDN	ISDN	3
3	USC†	B	4
4	RS-232	C	5

* The connector and port associated with paths 2.1, 2.2, 3, and 4 cannot be reconfigured.

† This connector can be converted to an X.21, V.35, V.36, RS-449, or RS-232 connector using cables. For more information, see your SuperStack II installation guide.

The port and path numbering conventions for built-in ISDN interfaces differ from other port and path types. For more information, see “Port and Path Numbering Issues for Built-in ISDN Interfaces” later in this chapter.

Before configuring ports and paths for the ISDN interface on your SuperStack II bridge/router, you must decide how you want to use the ISDN interface. For more information, see the Configuring Wide Area Networking Using ISDN chapter.

Port and Path Numbering on an OfficeConnect NETBuilder Bridge/Router

Table 5, Table 6, and Table 7 outline the default port and path numbering for model 11x, 12x, and 14x OfficeConnect NETBuilder bridge/routers.

Table 5 Path and Port Numbering for Model 11x OfficeConnect NETBuilder Bridge/Routers

Physical Path No.	Connector Mapped To	Logical Port No. Mapped To
1	10BASE-T or BNC (Depends on which connector is cabled.)	1
3	Flex WAN	4

Table 6 Path and Port Numbering for Model 12x OfficeConnect NETBuilder Bridge/Routers

Physical Path No.	Connector Mapped To	Logical Port No. Mapped To
1	10BASE-T or BNC (Depends on which connector is cabled.)	1
2	CSU/DSU	2

Table 6 Path and Port Numbering for Model 12x OfficeConnect NETBuilder Bridge/Routers

Physical Path No.	Connector Mapped To	Logical Port No. Mapped To
3	Flex WAN	4

Table 7 Path and Port Numbering for Model 14x OfficeConnect NETBuilder Bridge/Routers

Physical Path No.	Connector Mapped To	Logical Port No. Mapped To
1	10BASE-T or AUI (Depends on which connector is cabled.)	1
2.1	ISDN	2
2.2	ISDN	3
3	Flex WAN	4

The port and path numbering conventions for built-in ISDN interfaces differ from other port and path types. For more information, see “Port and Path Numbering Issues for Built-in ISDN Interfaces” later in this chapter.

Before configuring ports and paths for the ISDN interface on your OfficeConnect NETBuilder bridge/router, you must decide how you want to use the ISDN interface. For more information, see the Configuring Wide Area Networking Using ISDN chapter.

Configuring Multiple Paths to a Wide Area Port

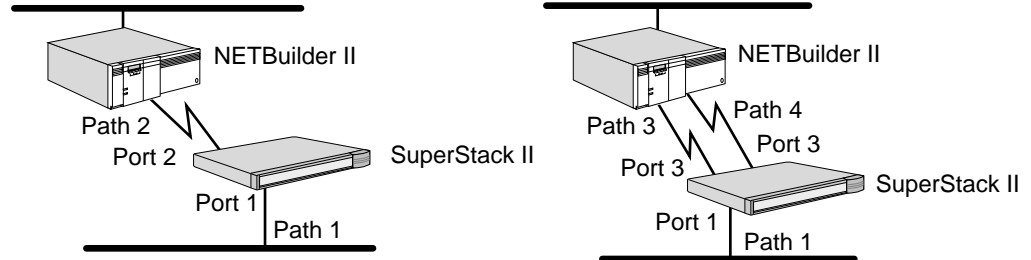
You can reconfigure the software so that multiple paths are mapped to one wide area port by entering the ADD -PORT PATHs command. If you assign multiple paths to a wide area port, the port must be running PPP. You can configure multiple paths to a wide area port for the following situations:

- If you have a WAN Extender attached to a NETBuilder II bridge/router, you can bind multiple WAN Extender virtual paths from the dial pool to a wide area port if the port is running PPP and Multilink Protocol.
- You can map multiple paths to one port to take advantage of the disaster recovery, bandwidth-on-demand features, and dial path pools if you are using ISDN or switch-56 circuit services directly or through a WAN Extender.

Figure 6 shows two sample topologies: the left topology has one path mapped to one port and the right topology has two paths mapped to one port (for disaster recovery or bandwidth-on-demand). In the left topology, path 2 is assigned by

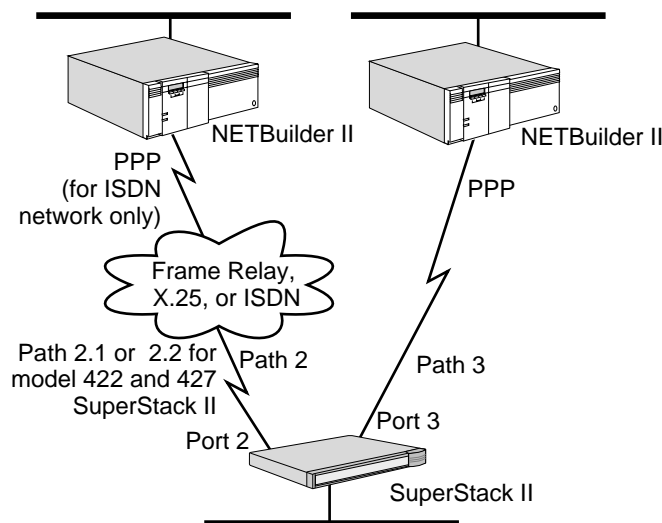
default to port 2. In the right topology, the software has been reconfigured so that paths 3 and 4 are mapped to port 3.

Figure 6 Possible Path-to-Port Assignments on a SuperStack II Bridge/Router



You can also assign one path to one wide area port and one path to another wide area port. In Figure 7, path 2 is assigned to port 2 and path 3 is assigned to port 3. Configure your wide area ports and paths as shown in this figure only if you plan to implement the Boundary Routing network resiliency feature on your SuperStack II NETBuilder bridge/router.

Figure 7 Setting Up Two Wide Area Ports on a SuperStack II Bridge/Router



Port and Path Numbering Issues for Built-in ISDN Interfaces

Built-in ISDN ports use a different path numbering convention from other paths. Each B channel is assigned a different path. For instance, 3.1 and 3.2 are path numbers for a built-in ISDN port, where 3 is the connector ID, and 1 and 2 are the channel IDs. Some commands require you to specify the connector ID and channel ID of an ISDN path.

If you do not specify a channel number for a parameter that requires it, the parameter is configured for channel 1 only. If you want to specify all channels associated with a physical interface, specify the connector number and an asterisk (for example, 2.*).

If you are unsure how to specify a path, see the description of the parameter in *Reference for Enterprise OS Software*.



When using an ISDN TA connected to a serial port, both B channels are assigned the same path number. You do not need to use the special ISDN syntax.

The syntax variation for these parameters is presented in *Reference for Enterprise OS Software* in the following format:

For non-ISDN interfaces

```
SETDefault !<path> -PATH remoteDialNo = "<string>"
SHow [!<path> | !*] -PATH remoteDialNo
SHowDefault [!<path> | !*] -PATH remoteDialNo
```

For built-in ISDN interfaces

```
SETDefault !<connectorID.channelID> -PATH remoteDialNo = "<string>"
SHow [!<connectorID.channelID> | !<connectorID>.*] -PATH remoteDialNo
SHowDefault [!<connectorID.channelID> | !<connectorID>.*] -PATH
remoteDialNo
```



Enterprise OS software menus and help strings do not display the syntax variation for ISDN interfaces.

For more information about configuring ISDN, see the Configuring Wide Area Networking Using ISDN chapter.

Configuring Local Area Interfaces

To set up ports and paths for local area interfaces on a bridge/router, follow these steps:

- 1 Assign a name to path 1 (optional) using:

```
SETDefault !<path> -PATH NAME = "string"
```

For example, to name path 1, enter:

```
SETDefault !1 -PATH NAME = "FLOOR_1"
```

Some restrictions apply to the name you assign using the -PATH NAME parameter. For more information, see the PATH Service Parameters chapter in *Reference for Enterprise OS Software*.

- 2 If you have a model 32x or 52x SuperStack II bridge/router, follow these steps:
 - a Configure the ring speed for the path using:

```
SETDefault !<port> -PATH BAud = 4000 | 16000
```

- b Enable the path using:

```
SETDefault !<path> -PATH CONTROL = ([Enabled | Disabled])
```

For example, to enable path 1, enter:

```
SETDefault !1 -PATH CONTROL = Enabled
```

- 3 Assign a name to port 1 (optional) using:

```
SETDefault !<port> -PORT NAME = "string"
```

Use a name that is easy to remember. For example, if port 1 is in Building 1, enter:

```
SETDefault !1 -PORT NAME = "BLDG_1"
```

Some restrictions apply to the name you assign using the -PORT NAME parameter. For more information, see the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

- 4 Disable all local area ports that you are not using by entering for each port:

```
SETDefault !<port> -PORT CONTROL = Disabled
```



3Com recommends that you disable all ports you do not use. Disabling unused ports improves bridge/router performance.

All ports are enabled by default. If the port has been disabled, you must re-enable the port to use it.

- 5 Repeat steps 1 through 4 for each local area port and local area path on your bridge/router.

This completes the setup procedure for local area ports and paths. To set up wide area ports and paths, go to the next section. To configure bridging or routing protocols, see the bridging and routing chapters in this guide.

Configuring Wide Area Interfaces

This section describes how to set up ports and paths for wide area interfaces on a bridge/router. To set up ports and paths for local area interfaces, follow the procedure described in "Configuring Local Area Interfaces" earlier in this chapter.



Before configuring ports and paths for the ISDN interface on SuperStack II bridge/routers with ISDN interfaces, you must decide how you want to use the ISDN interface. For more information about ISDN, see the Configuring Wide Area Networking Using ISDN chapter.

To set up ports and paths on a bridge/router with wide area interfaces, follow these steps:

- 1 Assign a name to the path (optional) using:

```
SETDefault !<path> -PATH Name = "string"
```

For example, to assign the path name SF-SJ, enter:

```
SETDefault !3 -PATH Name = "SF-SJ"
```

Some restrictions apply to the name you assign. For more information, see the PATH Service Parameters chapter in *Reference for Enterprise OS Software*.

- 2 If necessary, reconfigure the connector type for the path using the SETDefault -PATH CONNECTor command (the syntax varies depending on the platform you are using).

This step applies only to a NETBuilder II bridge/router with a high-speed serial (HSS) adapter card and to model 32x and 52x SuperStack II bridge/routers if you converted the serial connector labelled B (also referred to as the universal serial connector (USC)) to X.21, V.35, V.36, or RS-232 using a cable.

Table 8 summarizes which connector type to select if you have converted the serial connector labelled B on model 32x and 52x SuperStack II bridge/routers.

Table 8 Connector Setting for Converted Connectors on Model 32x and 52x SuperStack II Bridge/Routers

Connector Type Converted To	Setting of -PATH CONNECTor Parameter
X.21	X21
V.35	V35
V.36 or RS-449	RS449
RS-232	RS232



On a SuperStack II bridge/router with an RS-449 cable installed, the software cannot distinguish between the RS-449 cable and an V.35 cable. You must configure the -PATH CONNector parameter to RS-449; otherwise, the software assumes the cable is a V.35.

For the model 42x SuperStack II bridge/router, 3Com recommends retaining the default setting of the -PATH CONNector parameter (Auto). When this parameter is set to Auto, detection of the DTE connector type takes place when the platform boots.

For more information on the CONNector parameter and the auto startup feature, see the PATH Service Parameters chapter in *Reference for Enterprise OS Software* and to the Configuring Autostartup chapter in this guide, respectively.

- 3 If necessary, reconfigure the transmit clock setting for the serial path using:

```
SETDefault !<path> -PATH CLock = TestMode | External | Internal
```

The default setting is External. The bridge/router usually derives its clock from an external modem, so you do not need to change the External default setting.

If you have a model 32x or 52x SuperStack II bridge/router and you connected a serial connector to an IBM cluster controller, specify the Internal value.

If you have any other NETBuilder platform and you want the bridge/router to derive the clock from the onboard oscillator, specify the TestMode value. The TestMode value applies to all NETBuilder platforms *except* model 32x and 52x SuperStack II bridge/routers; the Internal value applies to model 32x and 52x SuperStack II bridge/routers only.

If you are configuring a NETBuilder II bridge/router with a WAN Extender, leave the transmit clock setting at External, the default.

You do not need to perform this step for the ISDN path for model 42x and 52x SuperStack II bridge/routers.



If you connect two NETBuilder II or SuperStack II bridge/routers to a NETBuilder II bridge/router with an HSS V.35 3-Port WAN interface, you must use a modem eliminator and set the CLock parameter to External on both devices. Contact your 3Com supplier for a suggested list of modem eliminators.

- 4 If necessary, reset the baud rate for the path using:

```
SETDefault !<path> -PATH BAud = <kbps>
```

For example, to set the baud rate of path 3 at 256 kbps, enter:

```
SETDefault !3 -PATH BAud = 256
```

For the default values and the range of baud rates available for each bridge/router platform, see the PATH Service Parameters chapter in *Reference for Enterprise OS Software*.



It is important to set the baud rate even if you use an external clock. The bridge/router uses the baud rate setting to allocate resources for a path, compute metrics, and select a forwarding path.

- 5 If the port is running PPP and you plan to use the features that use multiple paths mapped to a port (see "Configuring Multiple Paths to a Wide Area Port" earlier in this chapter), assign a path or multiple paths to each port using the ADD !<port> -PORT PAtHs command (the syntax varies depending on the platform you are using).

For example, to assign paths 3 and 4 to port 3, enter:

```
ADD !3 -PORT Paths 3,4
```

Assigning multiple paths to a port is supported only when PPP is the port owner.

To receive incoming calls from a remote site, you also can assign a dial path pool, WAN Extender dial-up lines, or WAN Extender channelized virtual paths to a port.

For example, the following command assigns all incoming calls from Boston using dial-up lines, WAN Extender dial-up lines, or channelized virtual paths to port 2:

```
ADD !2 -PORT Paths SCID "Boston"
```

If you use SCID to identify the incoming calls, you can only identify calls coming in from a 3Com NETBuilder bridge/router at the remote site.

If you are assigning WAN Extender virtual paths to be used over a T1 or E1 channelized leased line, you can avoid having to enter an SCID number by using the `-PORT WEProfileList` parameter. See this parameter in the PORT Service Parameters chapter in *Reference for Enterprise OS Software* for details. Using the `-PORT WEProfileList` parameter enables you to connect 3Com NETBuilder or other-vendor bridge/routers at the remote site to the central bridge/router port.

If you are assigning ISDN paths for the dial-up lines, you also have the option of using Calling Line Identification Presentation (CLIP) to identify the incoming calls from a 3Com NETBuilder or other-vendor bridge/router at the remote site to the central bridge/router port. CLIP is set up with the `-PORT CLList` and `-PORT DialRcvrState` parameters. See these parameters in the PORT Service Parameters chapter in *Reference for Enterprise OS Software* for details.

If a bridge/router virtual port is being configured for ISDN dial-up paths, modem dial-up paths, and for a leased line, the port should be configured for SCID and for CLIP to identify the port to the remote user. If a port is configured for both SCID and CLIP, the CLIP configuration will override the SCID configuration for incoming ISDN dial-up path calls. If the port is configured for something other than ISDN dial-up paths, SCID will be used and CLIP will be ignored.

For more information about setting up dial-up lines, see the Configuring Port Bandwidth Management chapter.

When you assign multiple paths to a port, a load-sharing algorithm is enabled. For more information, see "Load Sharing" in the Configuring Bridging chapter.

- 6 If you have previously disabled the path, changed the value of the `BAud`, `CLock`, or `CONNector` parameters, or assigned multiple paths to one port, re-enable the path using:

```
SETDefault !<path> -PATH CONTrol = Enabled.
```

If multiple paths are assigned to a port, you should enable all paths assigned to the port so that the load-sharing algorithm takes effect.

- 7 Repeat steps 1 through 6 for each wide area path you configure.
- 8 If necessary, enable or disable the wide area port.

All ports are enabled by default. If the port was previously disabled, you must re-enable it.

To enable or disable the port, use:

```
SETDefault !<port> -PORT CONTrol = Enabled | Disabled
```

- 9 Assign a name to the port (optional) using:

```
SETDefault !<port> NAmE = "string"
```

For example, to assign wide area port 3 the name SanJose, enter:

```
SETDefault !3 -PORT NAmE = "SanJose"
```

Some restrictions apply to the name you assign. For more information, see the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

- 10** If necessary, change the default owner of the wide area port using:

```
SETDefault !<port> -PORT OWNEr = ETHernet | TokenRing | FDDI | PPP |
FrameRelay | BSC | ATUN | SHDLC | SMDS | X25 | WanExtender | SDLC | ATM |
LoopBack | Auto
```

Table 9 lists the default port owner for each wide area port type.

Table 9 Default Port Owner for WAN Ports

Bridge/Router Model	Default Owner for WAN Ports
NETBuilder II bridge/router	PPP If an ATM module is installed, ATM is the default owner.
Model 2xx SuperStack II bridge/router	Auto
Model 32x SuperStack II bridge/router	Auto
Model 42x SuperStack II bridge/router	Auto for DTE serial ports; PPP for ISDN ports.
Model 52x SuperStack II bridge/router	Auto for serial ports; PPP for ISDN ports.

By default, the auto startup feature on the NETBuilder II and SuperStack II bridge/router can provide an automatic PPP or Frame Relay data link connection.

Auto startup does not provide an automatic SMDS, X25, WAN Extender, SDLC, or ATM data link connection. If the owner of the wide area port is one of these protocols, you need to manually set the value of this parameter to SMDS, X25, WanExtender, Synchronous Data Link Control (SDLC), or ATM as appropriate.

For complete information on the -PORT OWNEr parameter and the auto startup feature, see the PORT Service Parameters chapter in *Reference for Enterprise OS Software* and the Configuring Autostartup chapter in this guide, respectively.

For information on WAN Extender, see the *WAN Extender 2T/2E Installation Guide*, the *WAN Extender Manager User's Guide*, and the Configuring the NETBuilder II to use a WAN Extender chapter in this guide.

- 11** Repeat steps 8 through 10 for each wide area port you configure.
12 Disable each port that you are not using by entering:

```
SETDefault !<port> -PORT CONTrol = Disabled
```



3Com recommends that you disable all ports you do not use. Disabling unused ports improves bridge/router performance.

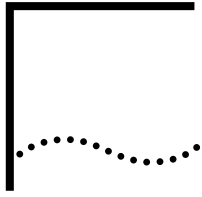
This completes the configuration for basic ports and paths. The new settings take effect immediately.

Many configurations and WAN technologies require the use of virtual ports. If you need to configure virtual ports, see the Configuring Advanced Ports and Paths chapter. To configure bridging or routing protocols, see the bridging and routing

chapters in this guide. Table 10 lists the primary bridging and routing protocols in this guide.

Table 10 Primary Bridging and Routing Chapters

If you want to configure:	See:
Bridging	the Configuring Bridging chapter
Source Route bridging	the Configuring Source Route Bridging chapter
AppleTalk routing	the Configuring AppleTalk Routing chapter
APPN routing	the Configuring APPN Intermediate Session Routing chapter
DECnet routing	the Configuring DECnet Routing chapter
IP routing	the Configuring IP Routing chapter
IPX routing	the Configuring IPX Routing chapter
OSI routing	the Configuring OSI Routing chapter
VINES routing	the Configuring VINES Routing chapter
XNS routing	the Configuring XNS Routing chapter



CONFIGURING ADVANCED PORTS AND PATHS

Many protocols and configurations require more ports than the basic port and path configurations provide. This chapter describes how to configure the following advanced port and path techniques:

- Virtual ports
- Multiple logical networks (port groups)

Virtual ports and port groups allow you to increase the port density on your bridge/router, and allow you to configure multiple logical ports that map to a single physical path.

For information on virtual ports, see “Using Virtual Ports” next. For information on multiple logical networks, see “Using Multiple Logical Networks” later in this chapter.

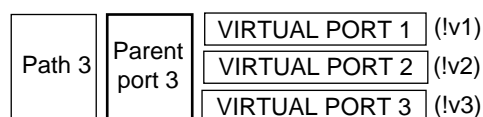
Using Virtual Ports

This section describes the concept of virtual ports, how to use virtual ports, and how to configure virtual ports. For information on virtual port concepts, see “Concept of Virtual Ports” next. For information on using virtual ports with wide area technologies, see “Parent Ports for X.25, PPP, Frame Relay, ATM and SMDS.” For information on using virtual ports for virtual LANs (VLANs) and Token ring In Fast Ethernet (TIFE), see “Virtual Ports for 802.1Q Virtual LANs.” For information on how to configure virtual ports, see “Configuring Virtual Ports.” For virtual ports with Layer 2 tunnelling, see the Configuring L2Tunnel Connections chapter.

Concept of Virtual Ports

You can configure multiple logical ports over one physical path on the platforms listed in Table 11. To configure multiple ports over one path, you create new logical interfaces called *virtual ports*. A virtual port is an object you define using software and associate with a nonvirtual port called the parent port (see Figure 8). A virtual port functions in the same way as a nonvirtual port, that is, as a logical interface that represents a connection to a network. The virtual port and its parent port share most of their properties. However, a virtual port and its parent port can be referenced separately by port-oriented software features, such as route policy and packet filtering, and can be distinguished by distinct wide area addresses.

Figure 8 Parent Port and Virtual Port



A virtual port can be connected to a network through a path providing a Frame Relay, ATM, or X.25 virtual circuit, or an SMDS Subscriber Network Interface (SNI).

A connection can also be made using PPP with dial-up features to achieve multidestination dialing (modem pooling and WAN Extender virtual path pooling).

Table 11 lists the bridge/routers that support virtual ports and the maximum number of virtual ports that can be configured on each bridge/router. There is no per-path limit, except that the total number of virtual ports configured on all paths cannot exceed the maximum for the bridge/router or switch.

Table 11 Bridge/Routers and Switches That Support Virtual Ports

Bridge/Router	Number of Virtual Ports Supported
NETBuilder II models, all software packages	512
SuperStack II NETBuilder SI 431, 432, 437, 438, 441, 442, 447, 448, 451, 452, 457, 458, 461, 462, 467, 468, 532, 537, 542, 547, 552, 557, 562, 567	48
SuperStack II model 327, 527	28*
OfficeConnect® NETBuilder model 111, 112, 113, 116, 117, 120, 121, 122, 123, 126, 127, 131, 132, 136, 137, 141, 142, 143, 146, 147	28
OfficeConnect NETBuilder model 100	10
PathBuilder S5xx series switches, all software packages	2048

* These platforms can act as a central node in a Boundary Routing topology.

Virtual ports function in the same way as nonvirtual ports. Table 1 provides information on topologies that require virtual ports and the node in the topology on which the virtual ports should be created.

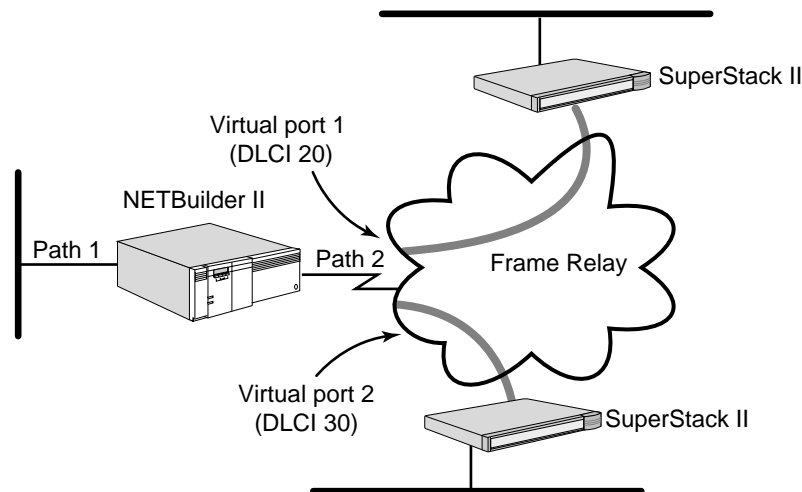
Table 1 Topologies that Require Virtual Ports

Topology	Virtual Ports Required?	Node That Virtual Ports Should Be Created On
Boundary Routing over Frame Relay, ATM DXI, or X.25	Yes.	Central node (NETBuilder II, SuperStack II model 227, 327, 427, or 527)
Traditional routed environment: partially meshed or nonmeshed Frame Relay, ATM DXI, and X.25 topologies	Depends on bridging or routing protocol. See "Virtual Ports over Frame Relay, ATM DXI, and X.25" later in this chapter for more information.	"Hub" router (NETBuilder II or SuperStack II model 222, 224, 227, 228, 327, 422, 424, 427, or 527)
Traditional routed or bridged environment: fully, partially, and nonmeshed ATM topologies	Yes.	NETBuilder II nodes on both ends of serial line running ATM
SMDS Service where there are more than 127 routers or more than one logical network segment (or 32 segments under IP), or a need to selectively filter packets among groups	Yes.	Depends on configuration
Multidestination dialing (modem pooling) over PPP	Yes, for dynamic dial-up lines.	Central node (NETBuilder II or SuperStack II Model 227, 327, 427, or 527)
Multidestination dialing (WAN Extender virtual path pooling) over PPP	Yes, for dynamic dial-up lines.	Central node (NETBuilder II only)

Table 1 Topologies that Require Virtual Ports (continued)

Topology	Virtual Ports Required?	Node That Virtual Ports Should Be Created On
Frame Relay topology with disaster recovery configured	Yes.	Nodes on both ends of serial line running Frame Relay
Routing traffic for 802.1Q virtual LANs.	Yes, to recognize VLANs. Traditional 802.1D bridging will forward VLAN packets without recognizing them as such.	Virtual ports required: Node that virtual ports should be created on: Node attached to LAN.

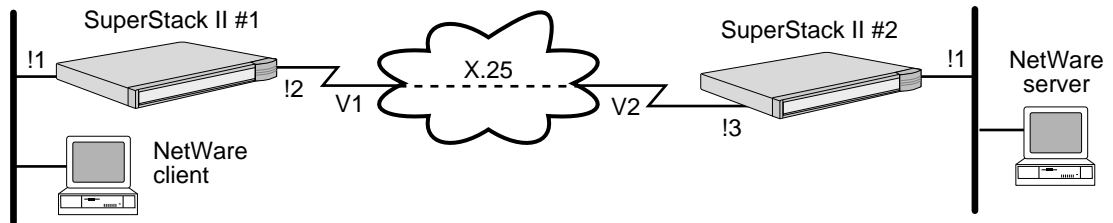
The sample Boundary Routing topology in Figure 9 demonstrates the use of virtual ports. This topology shows a NETBuilder II bridge/router with two paths labeled path 1 and path 2. Path 1 is an Ethernet interface. Path 2 is connected to a Frame Relay network that interconnects multiple local area networks through two SuperStack II boundary routers. Two virtual ports have been created on path 2. Each virtual port is a logical interface that represents a connection to one of the remote local area networks.

Figure 9 Topology Using Virtual Ports

Virtual Ports on SuperStack II Bridge/Routers

Virtual ports are numbered Vn , where n is a number from 1 through 28, which is the maximum supported on a SuperStack II bridge/router.

In Figure 10, virtual ports are configured on the wide area ports of two SuperStack II bridge/routers on both sides of an X.25 network. Each bridge/router has a virtual port defined over the path directly connected to the X.25 network. On SuperStack II bridge/router #1, virtual port V1 is defined over path 2. On SuperStack II bridge/router #2, virtual port V2 is defined over path 3.

Figure 10 Virtual Ports on a SuperStack II Bridge/Router

Virtual Ports and Different WAN Media

Virtual ports are used differently for different WAN media technologies. This section describes practical applications of virtual ports for different WAN media. For more information, see the following sections:

- "Virtual Ports over Frame Relay, ATM DXI, and X.25"
- "Virtual Ports over ATM"
- "Virtual Ports over SMDS"
- "Parent Ports for X.25, PPP, Frame Relay, ATM and SMDS"
- "Virtual Paths (WAN Extender only)"

Virtual Ports over Frame Relay, ATM DXI, and X.25

Frame Relay, ATM DXI, and X.25 are peer-to-peer protocols that connect two nodes on the network. Boundary Routing and bridging, Internet Protocol-Open Shortest Path First (IP-OSPF), DECnet IV, VINES, and Xerox Network Systems (XNS) require virtual ports because they do not provide a method for dealing with Frame Relay, ATM DXI, or X.25 topologies where bridge/routers are not directly connected to all others (full mesh). With Boundary Routing system architecture, when you create a virtual port over a particular path, each remote network attached to the Frame Relay, ATM DXI, or X.25 cloud is treated as a separate network.

Internet Protocol-Routing Information Protocol (IP-RIP), IP-Integrated Intermediate System-to-Intermediate System (IIS-IS) (NETBuilder II bridge/router only), Internetwork Packet Exchange (IPX), Intermediate System-to-Intermediate System (IS-IS), DECnet V, and AppleTalk can operate over partially meshed or nonmeshed Frame Relay, ATM DXI, or X.25 topologies without the use of virtual ports. The next-hop split horizon feature in IP-RIP, IPX, and AppleTalk allows communication between bridge/routers that are not directly connected to one another. To configure next-hop split horizon for these routing protocols, you must have a list of neighbors, which can be dynamically generated or manually configured in IP-RIP.

In IPX, you must manually configure neighbors for broadcast multiaccess (BMA) networks. For nonbroadcast multiaccess (NBMA) networks, for example, X.25 and Frame Relay, you can configure dynamic neighbor learning through the CONTROL parameter in the NRIP, SAP, and NLSP Services.

In AppleTalk, next-hop split horizon is configured by adding static mappings to the address mapping table.

You do not need to further configure IP-Integrated IS-IS and IS-IS to run over partially meshed or nonmeshed Frame Relay, ATM DXI, or X.25 topologies; you only need to configure neighbors.

Although it is not necessary to define virtual ports on IP-RIP, IPX, or AppleTalk routers in partially meshed or nonmeshed Frame Relay, ATM DXI, or X.25 topologies, virtual ports do provide the following additional benefits:

- A virtual port can be defined for each configured neighbor, which allows you to set up such features as filters and routing policies on a per-neighbor basis.
- Virtual ports provide greater control over your network.

If you want your NETBuilder II bridge/router or SuperStack II bridge/router to act as an Open System Interconnection (OSI) router in a Frame Relay, ATM DXI, or X.25 topology, you do not need to create virtual ports.

Table 2 lists each bridging and routing protocol and the technique you must use to deal with the lack of connectivity in partially meshed and nonmeshed Frame Relay, ATM DXI, and X.25 topologies.

Table 2 Connectivity in Partially Meshed and Nonmeshed Topologies

Protocol	Technique
Bridging	Virtual port
Boundary Routing	Virtual port
IP-RIP*	Next-hop split horizon
IP-OSPF	Virtual port
IP-Integrated IS-IS*	No special configuration required
IS-IS	No special configuration required
IPX*	Next-hop split horizon
APPN*†	No special configuration if sending APPN only over Frame Relay
DECnet IV	Virtual port
OSI/DECnet V	No special configuration required
VINES	Virtual port
XNS	Virtual port
AppleTalk*	Next-hop split horizon

* When configuring this protocol and another protocol that requires virtual ports over the same path, use virtual ports.

† The SuperStack II bridge/router does not support this protocol.

Virtual Ports over ATM

In an ATM environment, virtual ports are required in fully meshed and partially meshed topologies when bridging and routing. Nonmeshed topologies are supported, but they are not recommended.

Each ATM virtual port has a unique media access control (MAC) address.

Virtual Ports over PPP

PPP virtual ports differ from Frame Relay, ATM, X.25, and SMDS virtual ports in the following ways:

- A PPP virtual port can potentially use any path in the dial pool.
Frame Relay, ATM, X.25, and SMDS virtual ports are always associated with a particular path.
- PPP virtual ports do not have a parent port and operate independently. No parent port exists because the path was unbound from its port and placed into the dynamic dial path pool.
Frame Relay, ATM, X.25, and SMDS virtual ports inherit the attributes of the path over which they are defined. For more information, see “Parent Ports for X.25, PPP, Frame Relay, ATM and SMDS” later in this chapter.
- PPP virtual ports can be used with dial-up related parameters.
Frame Relay, ATM, X.25, and SMDS virtual ports cannot be used with dial-up related parameters.

You can use virtual ports and WAN Extender virtual paths in a PPP environment to provide dial pooling at the central site router. With dial pooling, a set of dynamic paths is unbound from their default ports and waits in the dial pool for an incoming call. When a call is received, the dynamic path that answers is assigned to a virtual port, which is standing by with the appropriate configuration information for the calling network. Because not all sites using a dial pool call the central site at the same time, it is possible to share a small group of paths with a larger group of sites. Each site that can potentially call into the dial pool has its own virtual port defined, so there are usually more virtual ports configured for the dial pool than dynamic paths assigned to it. For more information about WAN Extender virtual paths, see “Virtual Paths (WAN Extender only)” later in this chapter.

Virtual Ports over SMDS

Unlike Frame Relay, ATM, and X.25, SMDS provides a connectionless wide area network that also has multicast delivery capability, giving it LAN-like characteristics. Each attachment point to the SMDS network, the Subscriber Network Interface (SNI), can be assigned up to 16 individual addresses by the SMDS service provider. These addresses can be used to distinguish up to 16 distinct virtual SMDS ports over the same SNI. Unlike virtual ports for Frame Relay, ATM, or X.25, which connect to a single remote device, each virtual port in an SMDS environment connects to a distinct group of fully meshed devices. This connection allows the creation of a hierarchical, partially meshed structure that can exceed the SMDS address-screen-imposed limitation of 128 addresses in an SMDS network.

SMDS virtual ports provide additional points of control for configuring network and routing protocols, and for selectively applying port-level features such as filtering, route policy control, and route aggregation. Boundary Routing is not supported over SMDS.

Parent Ports for X.25, PPP, Frame Relay, ATM and SMDS

When you configure an X.25, Frame Relay, ATM, or SMDS virtual port, it inherits the attributes of the path over which it is defined. It also inherits some of the

attributes of its parent port. There are two kinds of inheritance: one is the inherited default for all VCs, and the other is when the port picks up the value of the parent port.

For PPP dial virtual ports, no parent port exists because the path was unbound from its port and placed into the dynamic dial path pool.

Unlike Frame Relay, ATM, X.25, and SMDS virtual ports, which are always associated with a particular path, PPP virtual ports can potentially use any path in the dynamic dial path pool. PPP virtual ports also can be used with dial-up related parameters.

For example, if you create a Frame Relay, ATM, X.25, or SMDS virtual port associated with a wide area port, the virtual port inherits port attributes from the following sources:

- Default and configured values of PORT Service parameters specified for a wide area port, with the exception of the following PORT Service parameters that are not related to X.25, Frame Relay, ATM, and SMDS virtual ports:

AutoDial	DialRetryCount
COMPRESSType	DialRetryTime
DialCONFig	DialSamplPeriod
DialCONTRol	DialSTatus
DialDebouncTime	LinkCompStat
DialHistory	OWNer
DialIdleTime	PAths
DialInitState	PathPreference
DialRcvrState	

The parameters in this list do apply to PPP virtual ports, such as SysCallerID virtual ports.

- Default and configured values of parameters from all other services specified for a wide area port.

To configure a virtual port, you must specify the virtual port and not the parent port. For example, if you are using the SETDefault !<port> -BCN CONTRol = Enabled syntax, you must specify the virtual port number instead of the parent port number for <port>.

Virtual Paths (WAN Extender only)

When you add a WAN Extender to a NETBuilder II bridge/router, it provides virtual paths that can be dynamically bound to a NETBuilder II physical or virtual port if PPP is running over the port. The NETBuilder II bridge/router can currently support up to 75 virtual paths. Because virtual paths are used by ports running PPP, multiple paths can be bound to a single port using the MultiLink Protocol.

Virtual paths can be used for WAN Extender ISDN and switch-56 dial-up lines and for WAN Extender T1 and E1 permanent leased channelized connections.

WAN Extender virtual paths are not bound to a port until a connection is established. While they are not bound, the virtual paths that are not configured to be used for channelized leased lines can serve as dynamic paths in a dial-up path

pool. The paths in the dial-up path pool are used for calls going through a port running PPP.

On ISDN or switch-56 dial-up lines, a virtual path binds to a port when an outgoing call is started or when an incoming call is received by the port. The virtual path goes back into the dial pool after the call is ended.

Like other dynamic paths, specific virtual paths in the dial-up pool can be dynamically bound to a port for bandwidth-on-demand or disaster recovery. After the demand and recovery is complete, the virtual paths unbind from their port or ports and return to the dial pool.

For channelized connections, such as T1 and E1, the virtual path binds to the port when the NETBuilder bridge/router and the WAN Extender synchronize with each other and the PPP negotiation is completed. The virtual paths used by channelized connections do not increase the number of paths in the dial pool after the call is ended.

Configuring Virtual Ports

This section explains how to configure virtual ports on your bridge/router. See Table 11 to determine whether your platform supports virtual ports.

Before setting up virtual ports for the ISDN interface on SuperStack II bridge/routers with an ISDN interface, you must decide how you want to use the ISDN interface.

Virtual Ports for 802.1Q Virtual LANs

In a virtual LAN environment, stations on a single physical bridged LAN are divided into logical groups, and are only allowed to communicate with other stations in that group. Each such subdivision is a virtual LAN (VLAN). The NETBuilder bridge/router implements standard IEEE 802.1Q VLANs and uses virtual ports to attach to VLANs.

When you create a VLAN virtual port, you specify the port that attaches to the physical LAN, and the 802.1Q vlan identifier. Routing protocols can then use this virtual port to route traffic for the member stations of that VLAN.

Each virtual port on a VLAN uses the same MAC addresses as the physical port.

Bridging between ports or virtual ports on the same physical LAN is not supported on the NETBuilder bridge/router.

A special use of VLANs as defined in 802.1Q is to tunnel token ring traffic over ethernet LANs. Other 3Com products see this as Token ring In Fast Ethernet, or TIFE. This type of traffic is useful in environments that are converting from token ring to ethernet LANs, or that support both.

The NETbuilder bridge/router supports TIFE traffic with VLAN tunnel virtual ports. Although these virtual ports connect to an Ethernet, they support source routing just like token ring ports.

Before configuring virtual ports, make sure that the owner of the wide area parent port is set appropriately.

To set up virtual ports, follow these steps:

- 1 Create a virtual port for each remote network that is attached to a Frame Relay, X.25, SMDS, ATM, or ISDN cloud, or that is running the PPP Protocol, using:

```
ADD !<port> -PORT VirtualPort {<path> {<FRDLCI> | <X.25 DTE> | SMDS |
    MPATM | ETHATM | TRATM | VlanTun <vld> | Vlan <vld>}} |
    SCID"<SysCallerID>" | | PPP | RAS | TunnelSwitch
```

Virtual ports are numbered Vn , where n is a number from 1 through the maximum supported on the bridge/router (see Table 11). You do not need to create virtual ports in numerical order. For instance, you can create virtual port V2 before V1.

For example, if you have a remote network on interface 1 that uses Frame Relay data link connection identifier (DLCI) 35, add virtual port V1 by entering:

```
ADD !V1 -PORT VirtualPort 1@35
```



ATM DXI ports also use the FR_DLCI value.

If you have a remote network on interface 3 that uses X.25 DTE 31107551234, add virtual port V3 by entering:

```
ADD !V3 -PORT VirtualPort 3#31107551234
```

If you have a remote network on interface 5 that uses SMDS, add virtual port V4 by entering:

```
ADD !V4 -PORT VirtualPort 5 SMDS
```

The command syntax for SMDS virtual ports does not use an individual DTE address. The virtual port does not take effect until its SMDSIndivAddr parameter has been configured.

If you have a remote network on interface 4 that uses multiprotocol encapsulation for ATM, add virtual port V5 by entering:

```
ADD !V5 -PORT VirtualPort 4 MPATM
```

If you have a remote network on interface 4 that uses LAN Emulation for ATM, add virtual port V5 by entering:

```
ADD !V5 -PORT VirtualPort 4 ETHATM
```

To create a PPP dial virtual port that uses the dynamic dial pool for its path resources in initiating and receiving calls from a remote router called NewYork, enter:

```
ADD !V3 -PORT VirtualPort SysCallerID"NewYork"
```

This command builds a mapping table entry between the virtual port and a remote bridge/router identifier (the site -SYS SysCallerID value of the remote NETBuilder bridge/router) and allows an incoming call to be mapped to a specific port. The remote router can be another NETBuilder II bridge/router or a SuperStack II bridge/router with an ISDN interface. Additional configuration steps are required to use the dial pool.

You can create multiple PPP virtual ports, but only one virtual port on a dynamic path can be active at a time.

To create a VLAN virtual port on port 1, for the vlan using 802.1Q identifier 42 (hex 3A), enter:

```
ADD !V9 -Port VirtualPort 1 Vlan 42
```

To create a VLAN virtual port for token ring tunneled traffic (TIFE) on port 2 with vlan id 7, enter:

```
ADD !V17 -PORT VirtualPort 2 VlanTun 7
```

- 2 If necessary, re-enable the virtual port.

Virtual ports are enabled by default. If virtual port V3 has been disabled, re-enable it by entering:

```
SETDefault !V3 -PORT CONTROL = Enabled
```

- 3 Assign a name to the virtual port (optional).

For example, to assign virtual port V3 the name First_St, enter:

```
SETDefault !V3 -PORT NAME = "First_St"
```

Some restrictions apply to the name you assign. For more information, see *Reference for Enterprise OS Software*.

- 4 Repeat steps 1 through 3 for each virtual port you configure.

This completes the configuration of virtual ports. The new settings take effect immediately. For information on Remote Access Services (RAS) virtual ports, see the Remote Access Services Parameters in *Reference for Enterprise OS Software*.

Using the Multiple Instance List

This section describes how to use the Multiple Instance List (MIL) feature on your bridge/router.

The MIL feature allows you to execute a command that can be applied to multiple instances at once, rather than applying the same command to each instance individually. An instance can be a port, path or slot number, a port or path name, or a group name.

Instances An instance is defined by '!' followed by a path name, port name, or group name. A MIL can be either several instances (separated by a space or by a comma (,)), or a range separated by a hyphen (-).

The command syntax for commands using MIL is:

```
command {!<instance>|!<MIL>} [arguments...]
```

The MIL syntax for instances is shown in Table 3.

Table 3 MIL Syntax for Instances and Their Meaning

Syntax	Meaning
!<instance> - !<instance>	Indicates a range of instances
!<instance> [,] !<instance> !<instance>	Indicates three instances
!<instance> [,] !<instance> - !<instance>	Indicates an instance and a range

Groups A group is a user-defined list of instances using the MIL format. For information on Remote Access Services (RAS) virtual ports, see the RAS Service Parameters chapter in *Reference for Enterprise OS Software*.

Defining a Group

You can define a group to cover instances that have common characteristics. The group name must be unique and cannot be '*'. The reserved group names include ALL, and SLOT1 through SLOT8. The reserved group name ALL would contain only present instances. A group cannot contain any other group as a member.

You can delete an entire group using the delete command, but you cannot delete a specific instance or instances from a group.

The parameter name for the group in the system services is InstanceGRoup.

The MIL syntax for group only is:

```
!InstanceGRoup <Group Name>
!InstanceGRoup <Group Name> [, ] !InstanceGRoup <Group Name>
```

The MIL syntax for instances and group is shown in Table 4.

Table 4 MIL Syntax for Instances and Group and Their Meaning

Syntax	Meaning
!InstanceGRoup [,] !<instance> - !<instance>	Indicates a group and a range
!InstanceGRoup [,] !instance>	Indicates a group and an instance

Using a Group

After you have defined a group, you can execute a command using the group name. If a group already exists, the instances will be added to that existing group. To add a group, use:

```
ADD [-SYS] InstanceGRoup <Group Name> "<Instance List>"
[ANY|Path\Port\Slot]
```

The group can be a port, path or slot. For example:

```
ADD -SYS InstanceGRoup gp3"!1, !3-!4
SHow -SYS InstanceGRoup gp3 Port !1, !3-!4
ADD -SYS InstanceGRoup gp3 "!5"
SHow -SYS InstanceGRoup gp3 Port !1,!3-!4, !5
```

The SHow command allows you to view all the available groups. You can also use the SHow command with the expand option to view all existing groups, including reserved groups (if there is a range specified), except for the type ANY. The expanded list of instances will not display repetitions of the instances.

For a description of the InstanceGroup parameter, see *Reference for Enterprise OS Software*.

Using Multiple Logical Networks

This section describes how to use and configure multiple logical networks on your bridge/router.

Concepts of Multiple Logical Networks

On the NETBuilder II bridge/router, the *multiple logical network* (MLN) feature allows you to:

- Group together multiple ports on a single bridge/router, and the LAN segments attached to them, to form a logical network. NETBuilder software can use groups in its network topology in the same way it uses virtual ports.
- Bridge network protocols, such as IP, among ports within a group.
- Route network protocols outside the group to other ports, virtual ports, or logical networks.
- Configure different MLN configurations for different protocols.
- Maintain configurations for protocols not configured for MLN, so that they bridge and route as usual, independent of the port groupings for other protocols.

Unlike conventional bridge/router operation, MLN provides simultaneous bridging and routing for the same network protocol. MLN enables you to integrate a number of bridged networks by routing from the bridged environments (configured as logical networks) across a LAN or WAN backbone. It also allows you to assign the same network number or subnet number to multiple physical paths. You can think of the logical network as a group of LAN segments that have been joined together to form a single network-level addressing domain.

When a conventional bridge/router is configured to bridge a particular protocol, all traffic for that protocol is bridged, and the router component is inactive, as shown in Figure 11. When it is configured to route that protocol, correctly addressed traffic for the protocol is routed, and the bridge component is inactive, as shown in Figure 12.



Bridging can occur even when the bridge/router is configured as a router. If a bridge/router receives packets of a protocol type that has not been configured on it, the bridge/router bridges the packets. If the -BRIDGE CONTROL parameter has been set to NoFireWall, incorrectly addressed routed packets are also bridged. The bridge/router can also be configured to bridge some protocols and route others. However, a conventional bridge/router without MLN cannot selectively bridge or route the same protocol, depending on destination.

Figure 11 Bridge/Router in Bridging Mode

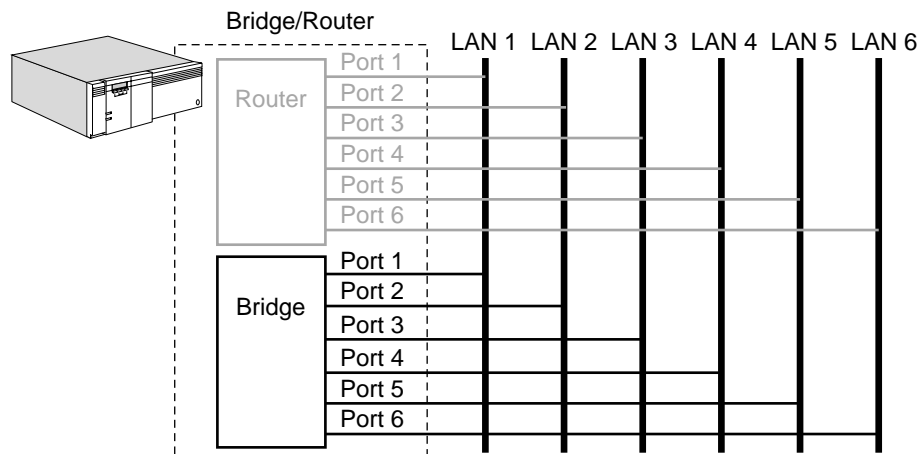


Figure 12 Bridge/Router in Routing Mode

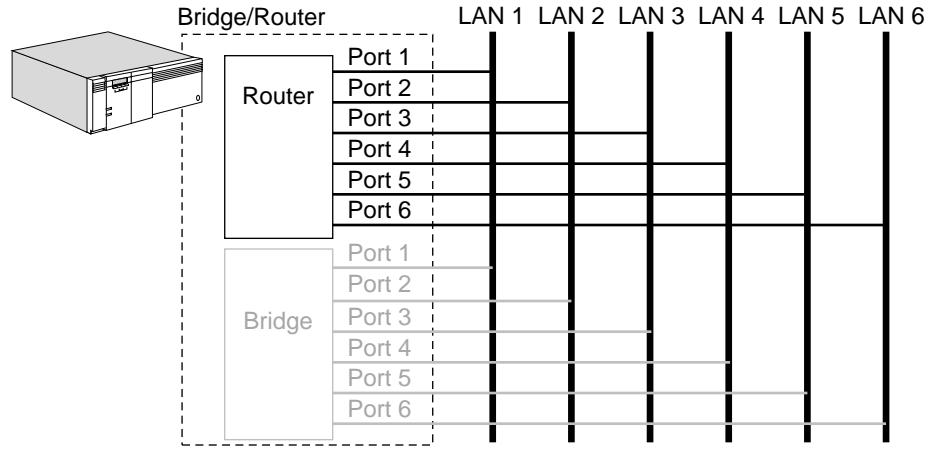


Figure 13 is an example of the simultaneous bridging and routing capability provided by MLN. Six networks are attached to a NETBuilder bridge/router. Each of the six networks has IP nodes, IPX nodes, and AppleTalk nodes. Ports 1, 2, and 3, and the LANs attached to them, have been grouped together into one logical network or port group, called V1. The logical interface between Enterprise OS software and this group is called a *group port*, and it is also identified as V1. The IP protocol has been configured on group port V1 (that is, V1 has been given an IP address). This IP address also applies to all ports in the group.

i *Group ports are numbered as if they were virtual ports.*

Ports 5 and 6, and the LANs attached to them, have been grouped into another logical network, V2. IP has also been configured on this group. IP has been configured individually on port 4, which has not been assigned to a group (that is, port 4 has been given an IP address).

Port groups have not been defined for IPX and AppleTalk. The bridge/router has been configured to route IPX. It has not been configured to route AppleTalk.

Figure 13 Multiple Logical Networks

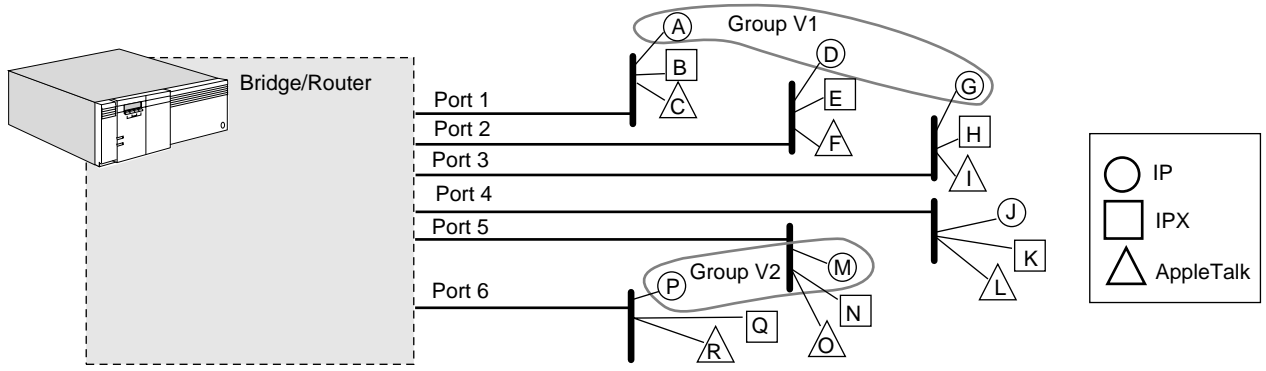


Figure 14 shows how IP traffic is handled in this configuration. IP is bridged among ports 1, 2, and 3 (as indicated in the figure by the MLN bridge, which is not a physical bridge but an internal software function). IP traffic is also bridged between ports 5 and 6. IP is routed between group V1 and all ports outside the

group, including port 4 and group port V2. IP is also routed between group V2 and all ports outside the group, including port 4 and group port V1.

Figure 14 IP Configuration Under MLN

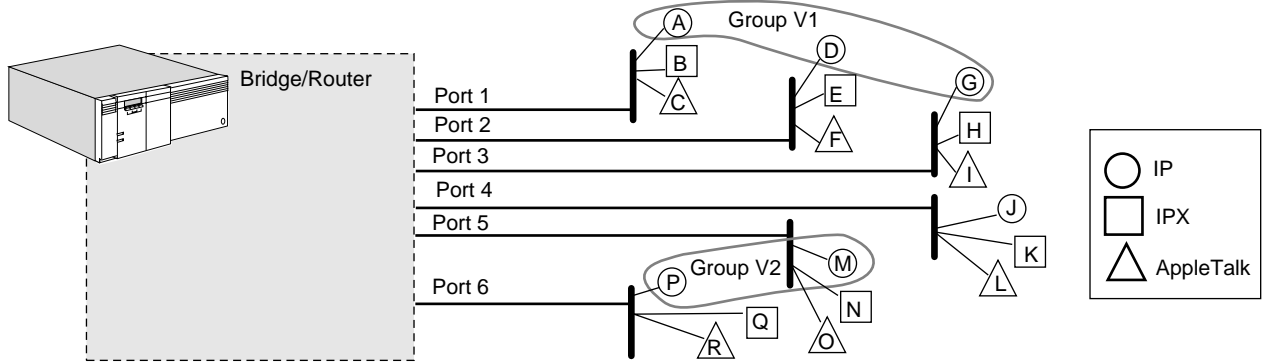
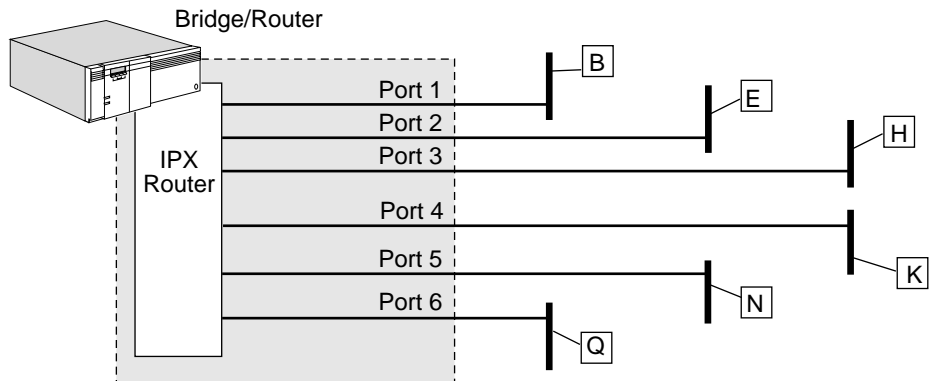


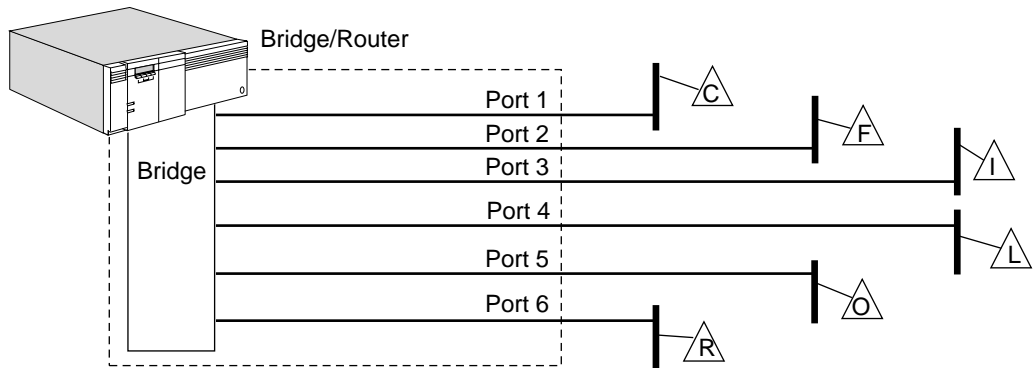
Figure 15 shows how the network looks to IPX. IPX traffic is routed among all ports, independent of the port groups defined for IP.

Figure 15 IPX Configuration Under MLN



AppleTalk routing is not enabled, so AppleTalk traffic is bridged among all six ports, as shown in Figure 16.

Figure 16 AppleTalk Configuration Under MLN



Only network protocols that configure a port group are affected by MLN. A protocol that does not participate in MLN can continue to configure its network topology at the port and virtual port level, including ports that belong to a port group for some other protocol. Bridged protocols such as NetBIOS and Logical Link Control, type 2 (LLC2) are also not affected by MLN.

MLN does not bridge between port groups, between a port group and a port, or between a port group and a virtual port. All of this type traffic is routed.

Software version 8.3 and later supports MLN for IP routing and transparent bridging over Ethernet. To configure logical networks, see the next section.

Configuring Multiple Logical Networks

This section describes how to set up MLNs by creating port groups and assigning ports to them. In software version 8.2 through version 9.1, you can create port groups only for Ethernet ports. Because version 8.2 through 9.1 supports MLN only for the IP protocol, create port groups only for ports over which you intend to route IP.

To create port groups, follow these steps:

- 1 To assign ports to a port group, use:

```
ADD !<port> -PORT LogicalNET ETHernet <port> [,...] ["<string>"] (1-50
characters]
```

where the first <port> is the group port that interfaces to the logical network. This port is always numbered as if it were a virtual port (Vn). The ports that follow the ETHernet parameter are assigned to the port group. These ports are called member ports, they cannot be virtual ports.

The last argument, " <string> ", which must be enclosed in quotation marks, is an optional descriptive name for the group port. It is displayed by entering the SHOW -PORT LogicalNET CONFIguration command.

For example, to add ports 1 and 2 to port group V1 enter:

```
ADD !V1 -PORT LogicalNET ETHernet 1,2 "Test Network B200 4th floor"
```

If port group V1 does not already exist, it is created and ports 1 and 2 are added to it. V1 also identifies the group port that references the group.

To add ports 3 and 4 to port group V2, enter:

```
ADD !V2 -PORT LogicalNET ETHernet 3,4
```

Port groups cannot overlap, that is, the same port cannot be configured as part of two different port groups.

- 2 If necessary, enable the group port.

Group ports are enabled by default. If group port V2 has been disabled, re-enable it by entering:

```
SETDefault !V2 -PORT CONTrol = Enabled
```

- 3 Assign a name to the group port (optional).

For example, to assign group port V2 the name " Bayfront, " enter:

```
SETDefault !V2 -PORT NAME = "Bayfront"
```

Some restrictions apply to the name you assign. For more information, see the -PORT NAME parameter in *Reference for Enterprise OS Software*.

- 4 Repeat steps 1 through 3 for each group port you configure.

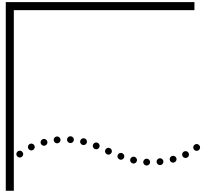
This completes the configuration of group ports. The new settings take effect immediately.



In addition to the CONTROL, NAME, and LogicalNET parameters, you can use the -PORT CONFIGuration parameter on group ports. To configure other port characteristics, configure them on member ports rather than the group port.

When you configure a logical network, you must enable global bridging and per-port transparent bridging on all member ports.

When a network routing protocol configures the group port in its network topology, it configures attributes for the entire port group.



CONFIGURING BRIDGING

This chapter describes how to set up, customize, and troubleshoot a bridge.

If you need to configure source route bridging, see the Configuring Source Route Bridging chapter.



For conceptual information, see "How the Bridge Works" later in this chapter.

Configuring Basic Bridging

This section describes how to set up a basic bridge. After you perform these steps, you can continue using the default values of parameters, or you can customize the bridge according to "Customizing the Bridge" later in this chapter.

Prerequisites

This section assumes that you have logged on with Network Manager privilege and set up ports and paths according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

To set up your bridge, first perform the procedure in "Transparent Bridging" next. You must then decide if any protocols that will be used on your network are source-route-only protocols; if they are, you may also need to perform the procedures described in the Configuring Source Route Bridging chapter. (To determine this, see the documentation that came with your protocol software.) If you want to set up your bridge as a wide area bridge, see "Bridging over a Wide Area Network" at the bottom of this page.

Transparent bridging is supported on Ethernet, token ring, Fiber Distributed Data Interface (FDDI), Point-to-Point Protocol (PPP), Frame Relay, Asynchronous Transfer Mode (ATM), X.25, Switched Multimegabit Data Service (SMDS), and Integrated Services Digital Network (ISDN).

Transparent Bridging

To set up a transparent bridge, enable bridging on the entire bridge/router by entering:

```
SETDefault -BRIDGE CONTROL = BRIDGE
```

If you do not want transparent bridging enabled on all ports, you can disable it on an individual port basis. For more information, see "Per-Port Transparent Bridging" later in this chapter.

Bridging over a Wide Area Network

You can set up your transparent bridge to forward packets over the following types of wide area networks:

- Frame Relay and Asynchronous Transfer Mode Data Exchange Interface (ATM DXI)
- X.25

- ATM
- SMDS
- PPP
- ISDN

Bridging over Frame Relay, ATM DXI, X.25, and ATM is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to bridge Frame Relay, ATM DXI, or X.25 over a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. When bridging over ATM in meshed, partially meshed, and nonmeshed topologies, you must create virtual ports. For configuration information, including a discussion of fully meshed, partially meshed, and nonmeshed topologies and virtual ports, see the *Configuring Wide Area Networking Using Frame Relay* chapter, the *Configuring Wide Area Networking Using the ATM DXI* chapter, the *Configuring Wide Area Networking Using X.25* chapter, and the *Configuring Internetworking Using ATM* chapter. For information on the number of virtual ports supported on each bridge/router platform, see Table 11 in the *Configuring Advanced Ports and Paths* chapter.

If you configure bridging over Frame Relay in a meshed topology with multiple data link connection identifier (DLCI) neighbors, the transmission of unknown unicast addresses and multicast (including broadcast) frames is processed separately from packets sent to known addresses. As a result, some frames may arrive at the destination nodes out of order.

For information on configuring transparent bridging over X.25 using X25User or X25DTE type profiles, see the *Configuring Wide Area Networking Using X.25* chapter in this guide and the *X.25 Service Parameters* chapter in *Reference for Enterprise OS Software*.

Bridging over SMDS is supported over a fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach groups of fully meshed devices). For more information, see the *Configuring Wide Area Networking Using SMDS* chapter.

To configure bridging over PPP, see the *Configuring Wide Area Networking Using PPP* chapter. For information about using the Spanning Tree Protocol over PPP, see “Using the Algorithm with Wide Area Bridges” and “Configuring the Spanning Tree Protocol over PPP” later in this chapter.

For more information on wide area networking using ISDN, see the *Configuring Wide Area Networking Using ISDN* chapter.

Bridging over Multiple Logical Networks

When you configure multiple logical networks (MLN), you must enable global transparent bridging by entering:

```
SETDefault -BRidge CONTrol = Bridge
```

If you do not enable global transparent bridging, MLN will be unable to bridge the configured network protocol, because stations on member ports will not be learned.



Over logical networks, Enterprise OS software supports only transparent bridging, not source route bridging. The only valid media type is Ethernet. For information about logical networks, see “Using Multiple Logical Networks” in the Configuring Advanced Ports and Paths chapter.

When you configure MLN, per-port bridging must remain enabled on all member ports. To display the per-port bridging configuration, use:

```
SHow !<port> -BRidge TransparentBRidge
```

If per-port bridging is disabled on any ports in a port group, re-enable it using:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

For complete information on the -BRidge TransparentBRidge parameter, see the BRidge Service Parameters chapter in *Reference for Enterprise OS Software*.

A bridge/router configured with MLN is normally connected to a backbone at the outside edge of a network. Consolidating networks into a logical network provides a desirable way to access a backbone: ports within the logical network are bridged, and access to the backbone is routed.

To provide further connectivity, you can also include bridges or switches within a logical network, as shown in Figure 17. Do not interconnect logical networks with bridges, or bridge between a logical network and another LAN segment. This topology defeats the MLN configuration and can cause your network to operate incorrectly.

Figure 17 Bridge within a Logical Network

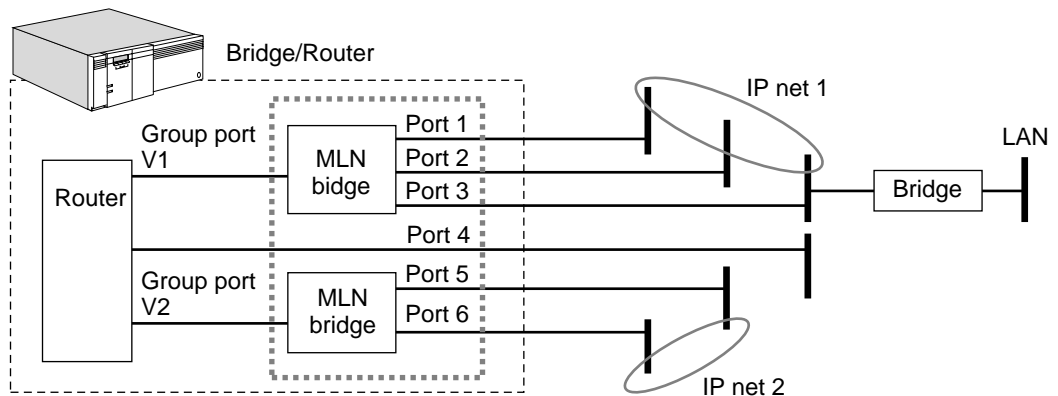
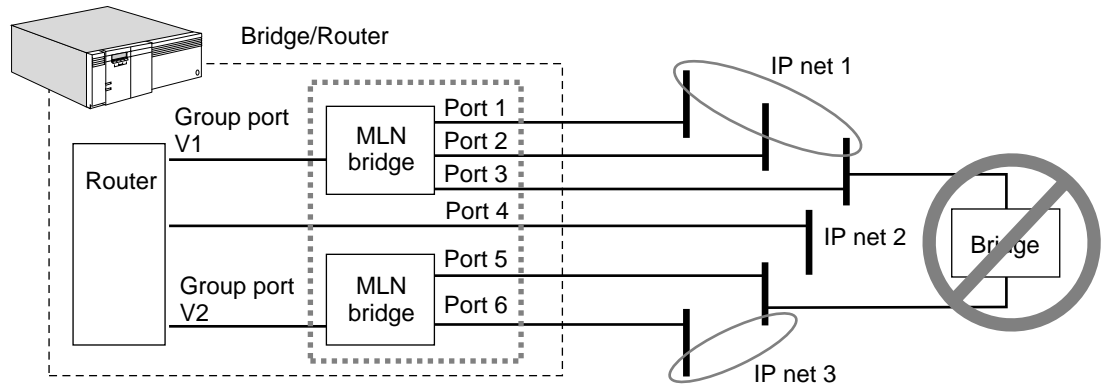


Figure 18 Bridge between Logical Networks

Configuring for Bridging and Routing

To configure your bridge/router, follow these steps:

- 1 Prepare your bridge/router for bridging.
- 2 Follow the instructions in the appropriate routing chapter to prepare your bridge/router for routing.

See Table 5 to find information on configuring specific protocols.

Table 5 Protocol Configuration

Protocol	Chapter
AppleTalk	15
APPN	11
DECnet	16
Internet Protocol (IP)	7
Internet Packet Exchange (IPX)	14
Open System Interconnection (OSI)	17
Vines	18
Xerox Network Systems (XNS)	19

You may need to see more than one chapter if your bridge/router will be used to route packets of different protocols.

- 3 Decide whether FireWall should be configured in the CONTROL parameter of the BRIDGE Service.

FireWall causes the bridge/router to discard all packets of a configured protocol that are addressed to destinations other than the bridge/router. There is a performance cost when FireWall is enabled, because every bridged packet must be checked. NoFireWall bypasses this check, resulting in better bridging performance.

NoFireWall is selected by default. To change the CONTROL parameter, use:

```
SETDefault -BRIDGE CONTROL = FireWall
```

or

```
SETDefault -BRIDGE CONTROL = NoFireWall
```


FireWall ensures that protocols configured for routing are never bridged by checking a type field in every packet to determine whether the packet should be routed or bridged. This level of checking slows down the performance of the bridge/router.

NoFireWall causes the router to skip the type field (except for broadcast packets, which are always checked). The bridge/router routes only data that is sent directly to the router at the MAC layer. All other packets are bridged. NoFireWall improves performance, but may forward incorrectly addressed packets. For example, if the IP protocol is configured for routing, and an IP packet is received with an address different from the router address, that IP packet is bridged.

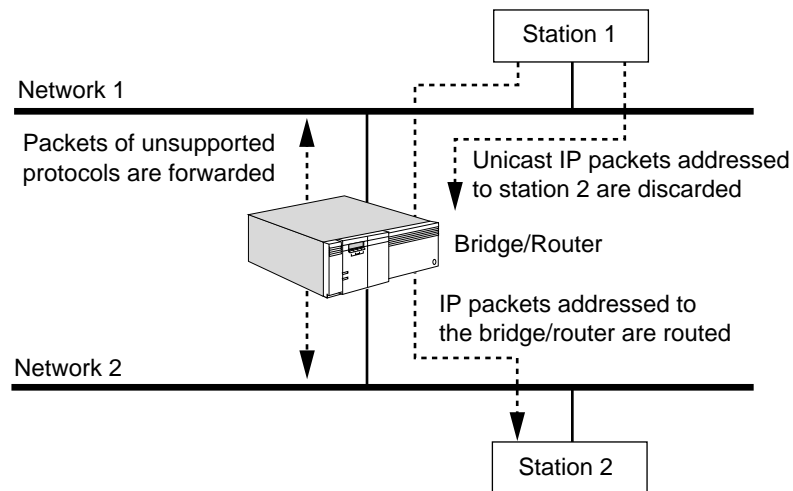
In Figure 19, the bridge/router performs IP routing and bridging of other protocols between networks 1 and 2. If station 1 wants to route a packet to station 2 via the bridge/router, the packet should be addressed to the bridge/router. If for some reason an IP packet is unicast to station 2, the FireWall value ensures that the bridge/router discards this packet.

If a bridge/router receives packets of a protocol type that has not been configured on it, it forwards the packets as if it were a bridge. In Figure 19, the bridge/router uses bridging to forward OSI packets between networks 1 and 2, because it is not configured for OSI routing.



Broadcast packets are not forwarded for any routed protocol, even if the -BRidge CONTrol parameter is set to NoFireWall. This procedure reduces unnecessary network traffic.

Figure 19 Effects of FireWall on Bridging and Routing



Verifying the Configuration

When you finish configuring your bridge, verify its configuration by following these steps:

- 1 Verify the values assigned to the -BRidge CONTrol parameter by entering:

```
SHoW -BRidge CONTrol
```

The current values of the -BRidge CONTrol parameter are displayed. FORWARD, LEarn, Aging, and NoFireWall, which are the default values, are usually selected. If the setting of any of these values has been changed, the bridge will perform one or more of the following processes:

- Not forward packets.
- Use only user-defined routes in its routing table to forward packets.
- Not check for addresses of nodes that appear to be “dormant.”
- Discard unicast packets of a protocol that is being routed (except for unicast packets to the bridge itself).

2 Verify that transparent bridging is enabled on the appropriate ports by entering:

SHoW -BRidge TransparentBRidge

The current values of the -BRidge TransparentBRidge parameter are displayed. Make sure that TransparentBRidge is selected on the appropriate ports.

3 Verify that the Spanning Tree Protocol is enabled by entering:

SHoW -STP CONTrol

The current values of the -STP CONTrol parameter are displayed. Make sure that Enabled is selected, which is the default value, to ensure that the bridge participates in the spanning tree network configuration.

4 Check the configuration of the transparent bridge and the status of each port and path by entering:

SHoW -BRidge CONFiguration

This is a sample display for a NETBuilder II 4-Slot chassis:

```
.....Current Configuration Values.....
CONTrol = (Aging,Bridge,NoFireWall,FOrward,LEarn)
AgeTime = 300

      Name      State      Type      Status      SRcSec      DStSec
      ----      -
Port 1  Port_1    Forwarding  TokenRing  Reachable   None       None
Path 1  Path_1
Port 2  Port_2    Forwarding  TokenRing  Reachable   None       None
Path 2  Path_2
Port 3  Port_3    Disabled   Remote     Unreachable Down       None
Path 3  Path_3    64 Kbps   Down (Wed Dec 31 16:00)
Port 4  Port_4    Disabled   Remote     Unreachable Down       None
Path 4  Path_4    64 Kbps   Down (Wed Dec 31 16:00)
```

The first two lines of this display show the current values of the -BRidge CONTrol and -BRidge AgeTime parameters. The remaining lines show the status of each port and its associated paths, including the name (if any) of the type of line, the time at which the paths status last changed, and the source and destination security.

5 If the display indicates that a port or a path is down, follow these steps:

a Check the configuration of each port by entering:

SHoW -PORT CONFiguration

b Check the configuration of each path by entering:

SHoW -PATH CONFiguration

6 Test the bridge by sending packets across it.

For example, make a connection from a device on one attached network to a host on another attached network. If you can successfully make a connection, the bridge is ready for normal operation. If you cannot make a connection, see “Troubleshooting the Configuration” later in this chapter.

Getting Statistics After your bridge is running, you may want to gather statistics.

You can collect statistics for a specified time period by using the `-SYS SampleTime` and `-SYS STATistics` parameters. For more information, see the *SYS Service Parameters* chapter in *Reference for Enterprise OS Software*.

For information on interpreting statistics displays, see the *Statistics Displays* appendix.

To gather statistics, follow these steps:

- 1 Display statistics for bridged packets by entering:

```
SHoW -SYS STATistics -BRidge
```

- 2 Display statistics for all ports by entering:

```
SHoW -SYS STATistics -PORT
```

- 3 Display statistics for all paths by entering:

```
SHoW -SYS STATistics -PATH
```

If the display indicates that there are errors on the attached network (for example, cyclic redundancy check errors), check the following items:

- The transceiver cable is properly attached to the transceiver.
- The transceiver is properly attached to the network cable.
- The network is properly terminated.

If the errors happen on a serial line, check the following items:

- Cable attachments
- Channel service units (CSUs) and digital service units (DSUs)
- Modems on each end of the serial line

If the line is a leased line, request help from the company that leases you the line (for example, the telephone company).

Troubleshooting the Configuration

To troubleshoot the bridge, follow these steps:

- 1 If one or more devices cannot communicate across the bridge, determine whether the filtering feature has been enabled and if so, what types of filters have been implemented.

- a Determine whether filtering has been enabled by entering:

```
SHoW -Filter CONTrol
```

- b If filtering has been enabled, review the filters by entering:

```
SHoW -Filter MASK  
SHoW -Filter POLicy
```

A filter defined incorrectly can cause packets destined for certain addresses to be discarded.

- c If filtering has been enabled, set up a counter to record the number of packets of a particular type that are forwarded by the bridge, using the `-Filter POLicy` parameter. For more information on bridge filtering, see the *Configuring Mnemonic Filtering* chapter.

- Determine if source and destination explicit forwarding have been implemented using:

```
SHow [!<port> | !*] -BRidge SRcSecurity  
SHow [!<port> | !*] -BRidge DStSecurity
```

- Check the routing table by entering:

```
SHow -BRidge AllRoutes
```

The address of the affected station should appear in the routing table, followed by the correct destination network number (port). If the address does not appear, make sure that the -BRidge CONTRol parameter settings include LEarn and FORward. If necessary, enter the ADD -BRidge ROUte command to add the address of the affected station.

- Display bridge configuration information and check the status of each path. Verify that each path is assigned to the appropriate network by entering:

```
SHow -BRidge CONFIguration
```

Check the physical attachments of any network not listed as REACHABLE or any path not listed as UP. Verify that the path is enabled by entering:

```
SHowDefault -PORT CONFIguration  
SHowDefault -PATH CONFIguration
```

- If the display indicates that a port or a path is down, follow these steps:

- Check the configuration of each port by entering:

```
SHow -PORT CONFIguration
```

- Check the configuration of each path by entering:

```
SHow -PATH CONFIguration
```

- Check for other activity on the bridge.

If there is no other activity, check the physical attachments of the bridge to its networks, including boards, back panel connectors, and transceiver or modem connectors. For lines to wide area bridges, check the CSU/DSU or modem and its configuration.

- If a large number of errors occur on the serial line of a wide area bridge to a remote network, check the physical lines. For a detailed account of errors on a given path, enter:

```
SHow -SYS STATistics -PATH
```

You can set some statistics to zero using:

```
FLush -SYS STATistics [-<service>]
```

- If a pair of devices cannot communicate across the bridge, check to see whether another pair can communicate across it.

If the second pair communicates, the problem is the first pair of devices, not the bridge.

To determine whether a pair of bridges can communicate with each other, initiate the data link test using the DLTest command. This test allows the bridges to exchange test packets and displays the resulting statistics. For more information on the DLTest command, see the Commands chapter in *Reference for Enterprise OS Software*.

- If possible, replace any bridge you suspect has a problem with another bridge or a repeater.

If the problem persists, the bridge is not the cause.

Customizing the Bridge

This section briefly describes the following features that enable you to customize your bridge:

- Per-port transparent bridging
- Static routes
- Bridge security
- Filters
- Translation bridging

This section discusses only the `DStSecurity`, `FunctionalAddr`, `MultiCastAddr`, `ROUte`, `SRcSecurity`, and `TransparentBRidge` parameters. For information on other `BRidge` Service and `STP` Service parameters, see the `BRidge` Service Parameters chapter and the `STP` Service Parameters chapter in *Reference for Enterprise OS Software*.

If you configure a logical network (port group), you can customize it by configuring global bridge properties, such as `AgeTime`, or individual properties of member ports, such as the ones described in this section. Configuring properties on the group port has no effect.

Per-Port Transparent Bridging

In addition to enabling transparent bridging on all ports (by setting the `-BRidge` `CONTRol` parameter to `Bridge`), you can enable transparent bridging on specified ports only using:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

To disable transparent bridging, use:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

For complete information on the `-BRidge` `TransparentBRidge` parameter, see the `BRidge` Service Parameters chapter in *Reference for Enterprise OS Software*.

Adding or Deleting Static Entries

To add a static (permanent) entry in the routing table used by your bridge, use:

```
ADD !<port> -BRidge ROUte <address>
```

To delete a routing table entry, use:

```
DELeTe [!<port>] -BRidge ROUte <address>
```



CAUTION: *When you change the owner for any WAN port, you must delete all static routes that were configured for the previous owner and WAN type. Use the `DELeTe -BRidge ROUte` command to delete these routes. Failing to delete the routes can cause a crash (fatal error) in the bridge/router software.*

For more information on the `-BRidge` `ROUte` command, see the `BRidge` Service Parameters chapter in *Reference for Enterprise OS Software*. For more information on routing tables, see “Routing Tables” later in this chapter.

Bridge Security

You can use bridge security features to select certain stations whose packets will be forwarded or blocked depending on their source or destination address. These features are applied only to packets traveling from one port of the bridge to another port. Packets addressed to the bridge are not affected.

The security features include:

- Source explicit forwarding.
- Source explicit blocking.
- Destination explicit forwarding.
- Destination explicit blocking.
- Combined source and destination security.



CAUTION: *Before you use the SRcSecurity and DStSecurity parameters, read the descriptions and examples in this chapter and the BRidge Service Parameters chapter in Reference for Enterprise OS Software. The SRcSecurity and DStSecurity parameters can affect bridge performance. Incorrect use of these parameters can cause the bridge to discard packets that you want to forward or to forward packets that you want to discard.*

For information on restricting packet movement based on packet contents, see “Filters” later in this chapter.

Source Explicit Forwarding

The Source Explicit Forwarding (SEF) feature allows you to forward packets from specific source addresses, on a per-port basis, in conjunction with a routing table.

To forward packets using the source explicit forwarding feature, you must enable forwarding on the port where the packet enters the bridge using:

```
SETDefault !<port> -BRidge SRcSecurity = Fwd
```

For a packet to be forwarded, its source address must be a static entry in the routing table on the port where the packet enters the bridge. Static entries are added or deleted using the ADD or DELETE -BRidge ROUTe <address> syntax. All other packets are discarded.



Some packets that meet forwarding conditions cannot be forwarded, because they are blocked by other constraints such as filtering or destination explicit blocking.

Figure 20(a) shows a bridge connecting two Ethernet networks, network A and network B. All stations on network B can communicate with all stations on network A. However, you can restrict packet forwarding so that only stations 1 and 2 on network A can communicate with the stations on network B. If stations 3 to 20 on network A send packets to any of the stations on network B, the packets are discarded.

To configure source explicit forwarding, follow these steps:

- 1 Set the -BRidge SRcSecurity parameter to Fwd on the port where the packet enters the bridge.

For example, to set this parameter to Fwd on port 1 of the bridge shown in Figure 20(a), enter:

```
SETDefault !1 -BRidge SRcSecurity = Fwd
```

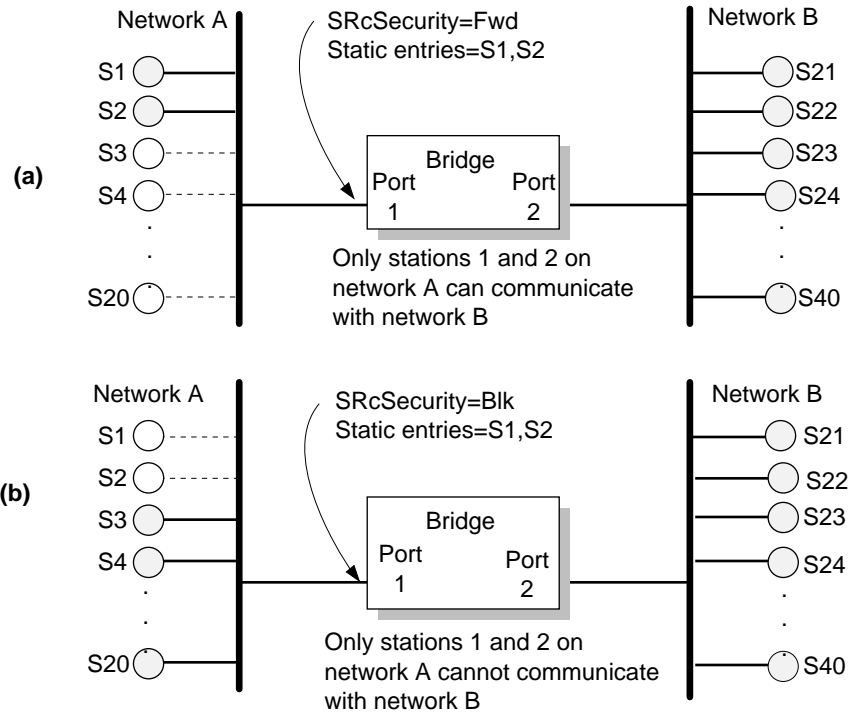
- 2 Add a static entry in the routing table for each source station from which you want packets to be forwarded.

The static entry must be generated on the port where the packet enters the bridge.

For example, to add a static entry for station 1 in Figure 20(a), enter:

```
ADD !1 -BRIDGE ROUTE %080002001234
```

Figure 20 Source Explicit Forwarding and Blocking



Source Explicit Blocking

The Source Explicit Blocking (SEB) feature allows you to discard packets from specific source addresses on a per-port basis in conjunction with a routing table. The blocking feature is the reverse of forwarding. Choose whichever feature (forwarding or blocking) allows you to enter fewer source addresses in the routing table.



On any port, only one of the two features, forwarding and blocking, can be turned on at a time.

To block packets using the source explicit blocking feature, you must enable the block feature on the port where the packet enters the bridge using:

```
SETDefault !<port> -BRIDGE SRcSecurity = Blk
```

For a packet to be blocked, its source address must be a static entry in the routing table on the port where the packet enters the bridge. Static entries are added or deleted using the ADD or DELETE -BRIDGE ROUTE <address> syntax. All other packets are forwarded (subject to other constraints).

Figure 20(b) shows a bridge connecting two Ethernet networks, network A and network B. All stations on network B must be able to communicate with all stations on network A. However, you can restrict packet forwarding so that if

stations 1 and 2 on network A send packets to the stations on network B, the packets are discarded. If stations 3 to 20 on network A send packets to the stations on network B, they are forwarded.

To set up source explicit blocking, follow these steps:

- 1 Set the `-BRidge SRcSecurity` parameter to `Blk` on the port where the packet enters the bridge.

For example, to set this parameter to `Blk` on port 1 of the bridge shown in Figure 20(b), enter:

```
SETDefault !1 -BRidge SRcSecurity = Blk
```

- 2 Add a static entry in the routing table for each source station from which you want packets to be blocked.

The static entry must be generated on the port where the packet enters the bridge.

For example, to add a static entry for station 1 as shown in Figure 20(b), enter:

```
ADD !1 -BRidge ROUTe %080002001234
```

If Source Explicit Forwarding were used in this example, the addresses of stations 3 to 20 would have to be manually entered in the routing table, requiring more work for the network manager.

Destination Explicit Forwarding

The Destination Explicit Forwarding (DEF) feature allows you to forward packets to specific destination addresses, on a per-port basis, in conjunction with a routing table.

To forward packets using this feature, you must enable the forward feature on the port where the packet enters the bridge using:

```
SETDefault !<port> -BRidge DStSecurity = Fwd
```

For a packet to be forwarded, its destination address must be a static entry in the routing table on the port where the packet exits the bridge. Static entries are added or deleted using the `ADD` or `DELeTe -BRidge ROUTe <address>` syntax. All other packets are discarded.

Figure 21(a) shows a bridge connecting two Ethernet networks in a company, network A and network B. All stations on network B must be able to communicate with all stations on network A. However, you can set DEF so that network A stations can send packets only to two stations on network B: station 21 and station 22. If stations on network A send packets to stations other than 21 or 22 on network B, the packets are discarded.

To configure destination explicit forwarding, follow these steps:

- 1 Set the `-BRidge DStSecurity` parameter to `Fwd` on the port where the packet enters the bridge.

For example, to set this parameter to `Fwd` on port 1 of the bridge shown in Figure 21(a), enter:

```
SETDefault !1 -BRidge DStSecurity = Fwd
```

- 2 Add a static entry in the routing table for each destination station that you want packets forwarded to.

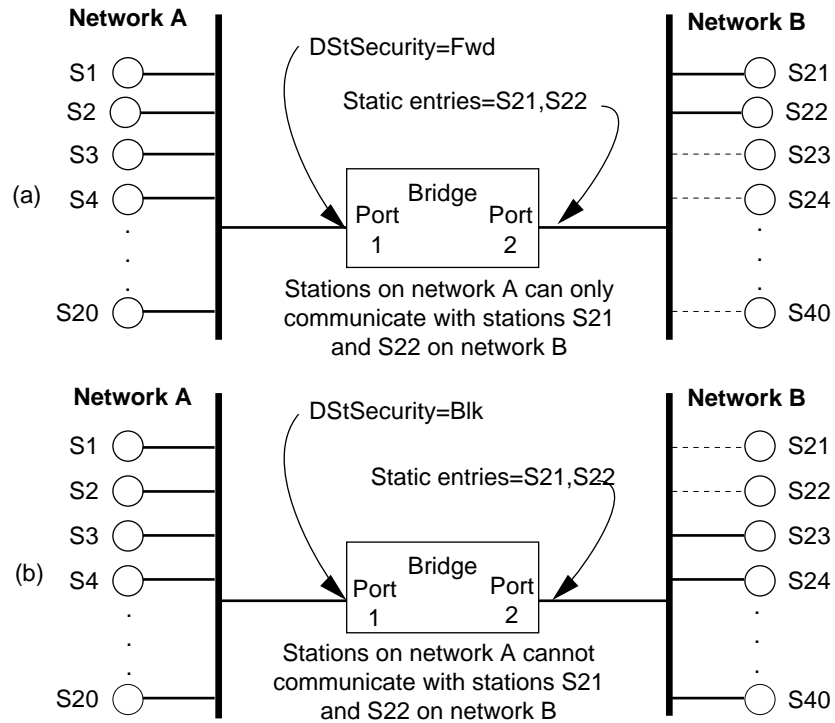
The static entry must be generated on the port where the packet exits the bridge. For example, to add a static entry for station 21 as shown in Figure 21(a), enter:

```
ADD !2 -BRIDGE ROUTE %080002001234
```



Some packets that meet forwarding conditions may not be forwarded, because they are blocked by other constraints such as filtering or source explicit blocking.

Figure 21 Destination Explicit Forwarding and Blocking



Destination Explicit Blocking

The Destination Explicit Blocking (DEB) feature allows you to discard packets sent to specific destination addresses, on a per-port basis, in conjunction with a routing table. The blocking feature is the reverse of forwarding. Choose whichever feature (forwarding or blocking) allows you to enter fewer source addresses in the routing table.



On any port, only one of the two features, forwarding and blocking, can be turned on at a time.

To block packets using this feature, you must enable the blocking feature on the port where the packet enters the bridge using:

```
SETDefault !<port> -BRIDGE DStSecurity = Blk
```

The destination address must be a static entry in the routing table on the port where the packet exits the bridge. Static entries are added or deleted using the ADD or DELETE -BRIDGE ROUTE <address> syntax. All other packets are forwarded (subject to other constraints).

Figure 21(b) shows a bridge connecting two Ethernet networks, network A and network B. Any station on network B must be able to communicate with any

station on network A. You can set Destination Explicit Blocking so that all stations on network A can communicate with the stations on network B except for stations 21 and 22. If stations on network A send packets to stations 21 or 22 on network B, the packets are discarded.

To configure destination explicit blocking, follow these steps:

- 1 Set the `-BRidge DStSecurity` parameter to `Blk` on the port where the packet enters the bridge.

For example, to set this parameter to `Blk` on port 1 of the bridge shown in Figure 21(b), enter:

```
SETDefault !1 -BRidge DStSecurity = Blk
```

- 2 Add a static entry in the routing table for each destination station that you do not want packets forwarded to.

The static entry must be generated on the port where the packet exits the bridge.

For example, to add a static entry for station 21 as shown in Figure 21(a), enter:

```
ADD !2 -BRidge ROUTe %080002001234
```

In this example, the `DStSecurity` parameter can be set to `Fwd`, but the addresses of stations 23 to 40 must be manually entered in the routing table, requiring more work for the network manager.

If you want the bridge to discard all packets destined for a particular address, use:

```
ADD -BRidge ROUTe <address> Off
```

Combined Source and Destination Security

You can build a complex bridge security system by combining `SEF`, `SEB`, `DEF`, and `DEB` features.

The `-BRidge SRcSecurity` and `-BRidge DStSecurity` commands can be turned on at the same port. However, if both forwarding and blocking conditions are met, the blocking condition takes precedence and the packet is discarded.

For example, suppose that the `SRcSecurity` parameter is set to `Fwd` and the `DStSecurity` parameter is set to `Blk`. A packet originating from an address that is not a static entry does not meet the forwarding condition. This packet is discarded regardless of its destination address.

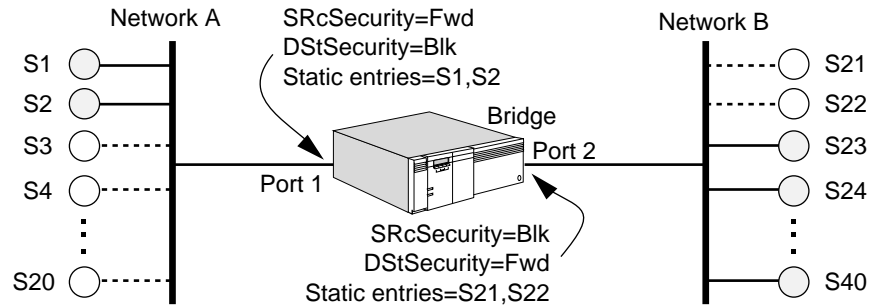
A packet originating from an address that is a static entry does meet the forwarding condition. However, if the destination address of this packet is also a static entry, then this packet meets the blocking condition and the packet is discarded.



Packets traveling on a network combining these features reach their destination only if all forwarding conditions are met and no blocking conditions are met.

Figure 22 shows a bridge connecting two Ethernet networks, network A and network B.

Figure 22 Combining Bridge Security Features



The network manager has set up the following complex system of bridge security features that restrict communication between the two networks:

- Stations 1 and 2 on network A *cannot* communicate with stations 21 and 22 on network B, because DStSecurity on port 1 is set to Blk and stations 21 and 22 are static entries on port 2. Although SRcSecurity is set to Fwd, and stations 1 and 2 on network A are static entries on port 1, the setting is ignored because blocking takes precedence over forwarding.
- Stations 1 and 2 *can* communicate with stations 23 to 40 on network B, because:
 - SRcSecurity is set to Fwd on port 1.
 - Stations 1 and 2 are static entries on port 1.
 - DStSecurity is set to Blk on port 1, but stations 23 to 40 are not static entries on port 2.
- Stations 3 to 20 on network A *cannot* communicate with any station on network B because:
 - SRcSecurity is set to Fwd on port 1.
 - None of the stations 3 to 20 are listed as static entries on port 1.
- Stations 21 and 22 on network B *cannot* communicate with any station on network A because:
 - SRcSecurity is set to Blk on port 2.
 - Stations 21 and 22 are static entries on port 2.
- Stations 23 to 40 on network B *can* communicate with stations 1 and 2 on network A because:
 - SRcSecurity is set to Blk on port 2 but stations 23 to 40 are not static entries on port 2.
 - DStSecurity is set to Fwd on port 2 and stations 1 and 2 on network A are static entries on port 1.

Filters You can use the -Filter MASK and -Filter POLicy parameters to define a custom filter so that packets meeting the criteria specified in the filter are forwarded or discarded. You can also restrict forwarding of packets by protocol type or other

packet contents. For complete information on configuring filters and policies, see the Configuring Mnemonic Filtering chapter.

Translation Bridging

Translation bridging is enabled by default, and there is a default table of functional-address-to-multicast-address mappings for well-known protocols. You do not need to take action unless you want to:

- Add an additional functional-address-to-multicast-address mapping to the default table.
- Use translation bridging across a serial line running PPP, Frame Relay, ATM DXI, X.25, or SMDS that connects a NETBuilder II bridge/router with an Ethernet module installed to a NETBuilder II bridge/router with a token ring module installed, as shown in Figure 23. If your network is configured in this way, you must be sure that the address resolution translation at each port that terminates the serial line is the same.

Figure 23 ARP Translation in a Bridged Environment

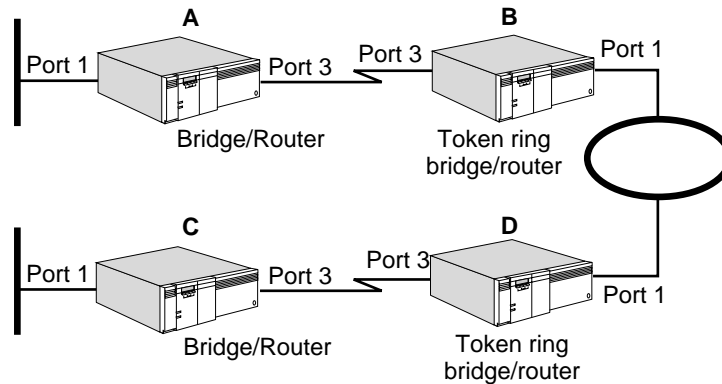


Table 6 lists protocol support in a mixed-media bridged environment. In this environment, all protocols are supported when one media type is communicating with a similar media type. 3Com does not support mapping of source-routed frames to transparent frames in a bridged environment except using source route transparent bridging gateway (SRTG) or Logical Link Control type 2 (LLC2) with data link switching. For more information, see the Configuring Source Route Bridging chapter and the Configuring Data Link Switching for SNA and NetBIOS Networks chapter. To be translated by the 3Com[®] implementation, all protocol packets must be transparent frames.

Table 6 3Com Protocol Support in a Translation Bridge Environment

Protocol	Ethernet to Token Ring End Station Connectivity	Routable Protocol
ARP	Yes	Yes
TCP/IP	Yes	Yes
XNS	No	Yes
IPX	No	Yes
AppleTalk	No	Yes
NetBEUI	Yes*	No

Table 6 3Com Protocol Support in a Translation Bridge Environment (continued)

Protocol	Ethernet to Token Ring End Station Connectivity	Routable Protocol
IBM NETBIOS	Yes*	No
3Com NBP	No†	No
LAT	Yes*	No

* For these protocols, the user must map the following multicast addresses to a unique functional address:
 IBM NETBIOS30000000001
 MS NetBeui30000000001
 LAT09002B0000F or 09002B020104

For LAT, officially assigned multicast-address-to-functional-address mappings should be added to the table with the -BRidge MultiCastAddr command. The address mapping should also be set on all bridges the protocol traverses.

† NBP Connection Request Datagram checksum is computed by NBP protocol on receipt to be different than the checksum value within the datagram; connection request is discarded.

Adding Functional-Address-to-Multicast-Address Mappings to the Default Table

This section describes how to add a functional-address-to-multicast-address mapping to the default table. For conceptual information, see "Address Mapping" later in this chapter.

To add a functional address, follow these steps:

- 1 Map a functional address to a multicast address using:

```
ADD -BRidge FunctionalAddr = %<address> MultiCastAddr = %<address>
```

- 2 Map a multicast address to a functional address using:

```
ADD -BRidge MultiCastAddr = %<address> FunctionalAddr = %<address>
```

For complete information on the -BRidge FunctionalAddr and -BRidge MultiCastAddr parameters, see the BRidge Service Parameters chapter in *Reference for Enterprise OS Software*.

There are 31 token ring functional addresses allowed, of which only 12 are user-configurable. Table 7 lists these user-configurable addresses. Some addresses may already be in use for other purposes.

Table 7 User-Configurable Addresses

Noncanonical Format	Canonical Format	Possible Other Use
C000 0010 0000	0300 0008 0000	AppleTalk ZIP/NBP
C000 0020 0000	0300 0004 0000	AppleTalk ZIP/NBP
C000 0040 0000	0300 0002 0000	AppleTalk ZIP/NBP
C000 0080 0000	0300 0001 0000	AppleTalk ZIP/NBP
C000 0100 0000	0300 8000 0000	AppleTalk ZIP/NBP, AMP discovery, DEC LAT
C000 0200 0000	0300 4000 0000	AppleTalk ZIP/NBP, DEC LAT
C000 0400 0000	0300 2000 0000	AppleTalk ZIP/NBP, DEC NetBIOS
C000 0800 0000	0300 1000 0000	AppleTalk ZIP/NBP, DECnet Phase IV
C000 1000 0000	0300 0800 0000	AppleTalk ZIP/NBP, DECnet Phase IV
C000 2000 0000	0300 0400 0000	AppleTalk ZIP/NBP

Table 7 User-Configurable Addresses (continued)

Noncanonical Format	Canonical Format	Possible Other Use
C000 4000 0000	0300 0200 0000	

Setting the Address Format

Protocol implementations on token ring can carry hardware MAC addresses within the protocol packets in either canonical or noncanonical format. End stations on Ethernet and FDDI always use canonical format. End stations on token ring can use either format. When a protocol implementation on token ring uses the noncanonical format within the protocol packet and is connected by a bridge to an Ethernet or FDDI LAN, then the interpretation of the MAC address becomes ambiguous, causing connectivity problems.

3Com has implemented the `-PORT ProtMacAddrFmt` parameter, which is user-configurable, to address the hardware ambiguity problem for the ARP protocol.

To set the address format, follow these steps:

- 1 Determine the address format that should be used by each port that terminates a serial line running PPP, Frame Relay, ATM DXI, X.25, or SMDS.

For example, in the configuration shown in Figure 23, you need to determine the MAC address format within the protocol packet that will be used by port 3 of bridge/routers A, B, C, and D.

For a complete description of the `-PORT ProtMacAddrFmt` parameter, see the `PORT Service Parameters` chapter in *Reference for Enterprise OS Software*. This description should help you decide whether the canonical or noncanonical format should be used on a particular port.

- 2 Set the address format on each port that terminates a serial line running PPP.

For example, in the configuration shown in Figure 23, to set the address format to noncanonical, enter:

```
SETDefault !3 -PORT ProtMacAddrFmt = NonCanonARP
```

in bridges A, B, C, and D.

Optimizing Bridge Performance

To improve the performance of the bridge, follow these steps:

- 1 Disable the firewall feature in mixed bridging and routing environments by entering:

```
SETDefault -BRIDGE CONTROL = NoFireWall
```

- 2 If the bridge is performing source route bridging, disable route discovery if the bridge does not need to send source route frames as an end station. Disable route discovery using:

```
SETDefault !<port> -SR RouteDiscovery = None
```

Setting this parameter to `None` means that the bridge transmits all end system packets as transparent frames, which can reach end systems in a transparent bridged or source route transparent (SRT) bridged environment.

- 3 Avoid configuring source and destination security features or filters.

- 4 After the bridge has learned addresses, disable dynamic learning (if you do not need it) by entering:

```
SETDefault -BRIDGE CONTROL = NoLEarn
```

- 5 If you do not need dynamic learning, increase the aging time for which entries remain in the routing table using:

```
SETDefault -BRIDGE AgeTime = <seconds> (10-1000000)
```

The default setting for this parameter is 300 seconds.

How the Bridge Works

This section provides conceptual information on the following topics:

- Transparent bridging
- Translation bridging
- Spanning tree algorithm
- Load sharing
- Routing tables
- Learning and filtering

Transparent Bridging

Transparent bridging is supported on Ethernet, token ring, FDDI, and the following wide area networks: Frame Relay, ATM, X.25, SMDS, PPP, and ISDN. When transparent bridging is enabled, the bridge forwards packets based on the destination address in the packets it receives. It also learns and records information about the location and addresses of devices on the surrounding networks, based on the source address in the received packets.

You can configure your bridge to forward frames using any of the following methods:

- Transparent bridging only
- Source route bridging only

The bridge forwards packets based on a route determined by the source or end system from which the packet originated. Because the end system and not the bridge determines the route, a bridge using source route bridging does not record or learn information about addresses on the surrounding networks in the way that a transparent bridge does.

- Transparent and source route bridging simultaneously

Operating transparent and source route bridging simultaneously is called source route transparent bridging. The bridge automatically determines whether a packet should be forwarded using transparent bridging or source route bridging.

When configuring parallel bridges, 3Com recommends that you configure both bridges in the same mode, either source route or source route transparent, to prevent unexpected blocking of one type of traffic. For more information on source route bridging, see the Configuring Source Route Bridging chapter.

- Source route transparent bridging gateway (SRTG)

You can connect source route domains to transparent bridging domains by configuring SRTG. The SRTG software provides a mapping between the two domains, so that a user on a token ring network using source routing can

communicate with another user on an Ethernet network using transparent bridging. For more information, see the Configuring Source Route Bridging chapter.

IBM-Related Services

IBM-related services such as data link switching (DLSw) and APPN are affected by parameter settings in the BRIDGE, SR, and LLC2 Services. Figure 24 lists the required settings in source route (SR), source route transparent (SRT), and transparent (T) bridging environments for each of the IBM-related services.

In this table, the bridging environment (SR, SRT, or T) is shown in the Port Configuration column. Tunneling is the 3Com proprietary method of LLC2 tunneling, DLSw is data link switching, and LNM is LAN Net Manager. The settings are shown in abbreviated form. For example, the row labeled DLSw/Tunneling with port configuration SR represents DLSw or 3Com LLC2 tunneling in a source-route-only port configuration. The entries in this row expand to the following software configuration commands:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
SETDefault !<port> -BRidge TransparentBridge = NoTransparentBridge
SETDefault -BRidge CONTrol = Bridge | NoBridge
SETDefault !<port> -SR RouteDiscovery = LLC2
SETDefault !<port> -LLC2 CONTrol = Enable
SETDefault !<port> -SR RingNumber = <number> (1-4095) |
0x<number>(1-FFF)
```

In this configuration, global bridging is enabled or disabled on one or more token ring ports. Transparent bridging is disabled, source routing and route discovery are configured, and LLC2 is enabled.

Figure 24 IBM-Related Feature Settings

Services	Port Configuration	Source Route Bridging (-SR SRB)	Transparent Bridging (-BR TBR)	Bridging (-BR CONT)	Route Discovery (-SR RD)	LLC2 CONTROL (-LLC2 CONT)	Frame Copy Errors
Bridging only	SR	SRB	NTB	BR	NoLLC2	Disable	None
Bridging only	SRT	SRB	TB	BR	NoLLC2	Disable	*
Bridging only	T	NSRB	TB	BR	NoLLC2	Disable	*
LNМ	SR	SRB	NTB	BR	LLC2	Enable	None
DLSw/ Tunneling	SR	SRB	NTB	BR NBR	LLC2	Enable	None
DLSw/ Tunneling	SRT	SRB	TB	BR	LLC2	Enable	* †
DLSw/ Tunneling	T	NSRB	TB	BR	NoLLC2	Enable	* †
APPN	SR	SRB	NTB	BR NBR	LLC2	Disable	None
APPN	SRT	SRB	TB	BR NBR	LLC2	Disable	*
APPN	T	NSRB	TB	BR NBR	LLC2	Disable	*
Default Setting	SRT	SRB	TB	NBR	NoLLC2	Disable	None

* In this configuration, end systems may generate a small number of MAC Frame Copy error report packets when the bridge/router is initializing or when it ages out a MAC address from its bridge table.

† In this configuration, it is important for global bridging to be enabled, otherwise, the token ring hardware does not filter transparent packets. This can generate many Frame Copy error reports and adversely effect performance.

Token Ring Frame Copy Errors

For transparent bridge (TB) or SRT configurations, token ring end systems may generate a small number of MAC Frame Copy error reports when a NETBuilder II bridge/router is initializing or when the bridge/router ages out a MAC address from its bridge table.

For the bridge/router to learn the MAC addresses of transparent end systems on the token ring, it copies a packet with an unknown source address and sets the address-recognized (A) and frame-copied (C) bits in the Frame Status (FS) field. A problem occurs when the FS (A) and (C) bits have been set and the destination of the frame is an end system on the local ring. The destination end system expects the (A) and (C) bits to be zeros. When it receives a frame with these values already set, it reports an error. The end system counts these errors until the error threshold is reached; then it sends out a MAC Report Error packet.

These Frame Copy errors occur only with transparent token ring packets, because the bridge/router hardware filters source-routed packets based on the route information field, not the MAC address. If the bridge/router is configured for source route only, it never copies frames destined for a station on the local ring. These errors can be avoided by running in source-route-only mode.

Figure 24 identifies those configurations that can cause Frame Copy errors.

Translation Bridging

With translation bridging, you can communicate between transparent bridging end stations on different LAN media types: Ethernet, token ring, and FDDI. (For source route end stations to communicate with transparent bridging end stations, you

must use SRTG as described in the Configuring Source Route Bridging chapter.) You also can communicate between end stations on the same media type across backbones of a different media type. The 3Com implementation of translation bridging is based on general principles of media access control (MAC) header translation and encapsulation, as well as protocol-specific translation for well-known protocol problems.

When a packet needs to be forwarded from a token ring or FDDI network to an Ethernet network, translation bridging transforms the packet from a token ring or FDDI format to an Ethernet format, or vice versa. When a packet is forwarded to a serial port, translation bridging takes place automatically at the remote bridge port when it receives the packets. For translation bridging to occur on wide area bridges, translation software is necessary in both units.

Translation bridging between Ethernet and token ring networks connected by a NETBuilder II bridge/router can take place either across serial lines or through a local port. Translation bridging between Ethernet and FDDI networks takes place through local ports. Figure 25 illustrates each of these concepts.

Figure 25 Using Translation Bridging to Interconnect Token Ring and Ethernet Networks

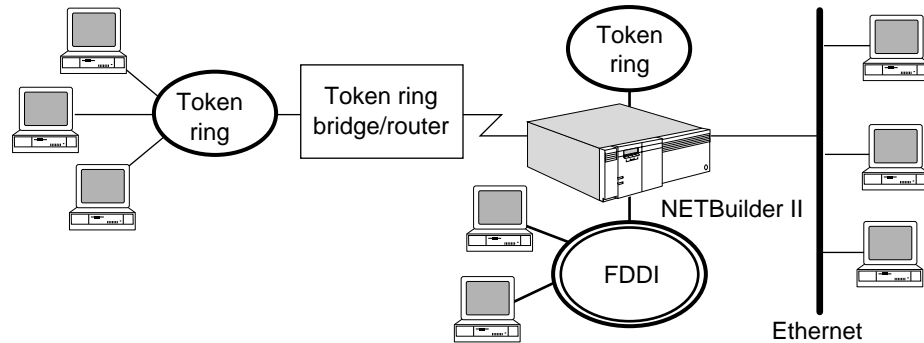
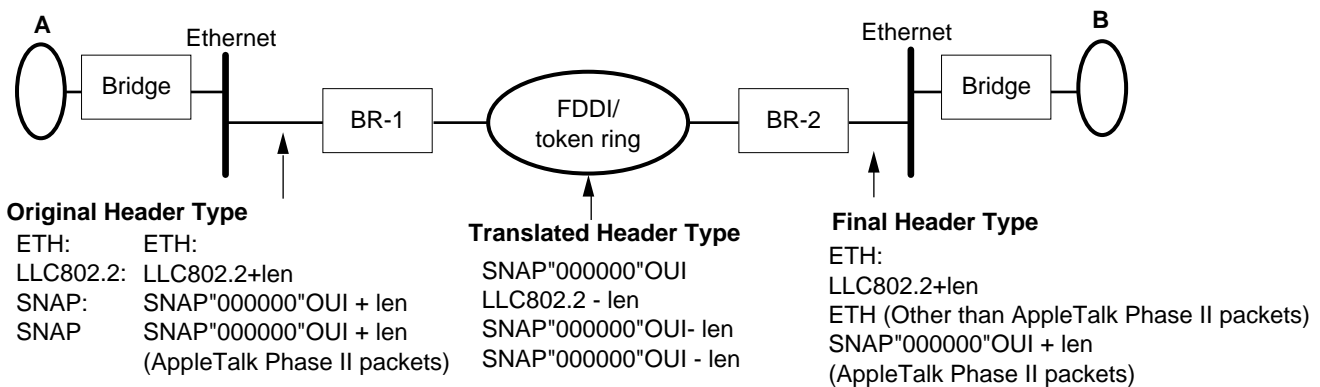


Figure 26 illustrates the general principles of MAC header translation, as applied to bridging packets of different formats on Ethernet over a token ring or FDDI backbone.

Figure 26 MAC Header Translation



OUI Packets

AppleTalk Phase 2 packets originating on an Ethernet network and destined for another Ethernet network across an FDDI backbone remain in AppleTalk Phase 2 Subnetwork Access Protocol (SNAP) format. SNAP packets use an Organizationally Unique Identifier (OUI) of 000000. AppleTalk Phase 1 packets originating on an Ethernet network are tunneled through the FDDI backbone using an OUI value of 0000F8.

For networks other than AppleTalk Phase 2, a SNAP header on Ethernet is translated to a SNAP header on FDDI and then back to Ethernet, instead of SNAP. If you are using translation bridging from Ethernet to Ethernet across FDDI, use the Ethernet header format on both sides.

Some protocols use a format similar to SNAP for encapsulating other types, but use their own proprietary OUI instead of the 000000 OUI used by SNAP. These packets are not converted back to Ethernet when bridging from FDDI or token ring onto an Ethernet LAN.

Maximum Transmission Unit

The maximum transmission unit (MTU) is the maximum packet size allowed, which varies according to the LAN media. Applications that run in a multimedia bridged environment must be configured to use packet sizes that are equal to or smaller than the smallest of the MTU sizes in the extended LAN; otherwise, some media may drop oversized packets. If a particular application cannot accept smaller packets, using network layer routing instead of MAC layer bridging may provide a solution.

For IP packets being bridged between interfaces that have mismatched MTU sizes, you can enable the IP fragmentation feature by setting the `-BRIDGE CONTROL` parameter to `IPFragment`. The bridge then fragments IP packets that are being forwarded to ports with a smaller MTU size.

LLC Length and Packet Size

LLC packets on Ethernet networks contain a length field that is removed before the packet is transmitted to FDDI and token ring media. In some systems, the actual length of the packet and the LLC length field may not match. When these packets are bridged to another Ethernet across an FDDI or token ring backbone, the resulting packet length cannot be determined. The 3Com implementation ignores the actual packet length and transmits according to the LLC length field. If the actual length of the packet is greater than the LLC length, it is cut short to correspond to the LLC length. If it is less than the LLC length, the packet is padded at the end to match the LLC length.

Address Mapping

On Ethernet and FDDI media, multicast addresses are used in the destination address field to reach a group of stations running a certain type of protocol. Because the multicast address is identified by one bit in the address space, it is possible to have millions of such addresses in the available 48-bit address space.

For similar applications on token ring media, functional addresses are used. Only 32 functional addresses are possible. When bridging packets from Ethernet or FDDI to token ring, multicast packets should be mapped to the corresponding

functional address and vice versa. Multicast packets that do not have a one-to-one mapping are dropped.

The 3Com implementation maintains a table of multicast-to-functional address mappings for well known protocols, shown in Table 8. User-defined mappings can be added using the `-BRidge MultiCastAddr` or `FunctionalAddr` parameters. For further information, see “Adding Functional-Address-to-Multicast-Address Mappings to the Default Table” earlier in this chapter.

Table 8 Multicast-to-Functional Address Mappings

Type of Packet	Token Ring Functional Address	FDDI or Ethernet Multicast Address
Broadcast	0300FFFFFFFF	FFFFFFFFFFFF
Broadcast	FFFFFFFFFFFF	FFFFFFFFFFFF



By default, the bridge displays addresses in canonical format.

Priority Mapping

Token ring and FDDI media provide a means of prioritizing access over the ring. Applications can request a priority between 0 and 7, and the MAC sublayer maps these user priorities to access priorities supported by the individual media access methods. A token with an access priority equal to or less than the requested user priority transmits this packet over the media.

To prevent a bridge from reordering frames of a given user priority received on one port when forwarding to another port, user priority information is conveyed to the driver along with the frames submitted for transmission. The mapping of user and access priorities is done in accordance with the 802.1d IEEE standard for MAC bridging. For packets that are bridged from Ethernet to token ring, the default user priority of 4 is used.

Configuring Address Format

If you are connecting a 3Com bridge to a bridge from another vendor and are bridging token ring or FDDI packets over a WAN link, you can configure the `DatalinkAddrFmt` parameter to ensure that the 3Com bridge conforms to standards used by the other bridge.

Protocol-Specific Issues

The following section describes protocol-specific translation bridge issues.

AppleTalk 3Com has not implemented translation bridging of AppleTalk packets between token ring and other media. Communication between AppleTalk nodes on token ring and nodes on other media types should be accomplished using routing.

Bridging of AppleTalk Phase 1 and Phase 2 packets between Ethernets across an FDDI backbone is implemented according to the recommended practice published by IEEE. This bridging is controlled by the `-BRidge APPLetalk` parameter. 3Com recommends that you retain the default value of Enable.

IP MAC-layer bridging typically does not bridge large frames between physical media that have dissimilar maximum frame sizes. To solve this problem, 3Com implements fragmentation of bridged IP packets. IP fragmentation is supported

between LAN media and also on WAN media. Fragmentation can be enabled by setting the -BRIDGE CONTROL parameter to IPFragment.



Fragmentation may cause some deterioration in performance.

IPX 3Com supports IPX translation bridging between end stations on the same LAN media type across a backbone of another media type. Bridging IPX between end stations on different media types is not supported.

NetWare stations running IPX can be configured to operate in pure Ethernet format, SNAP format, or 802.2 LLC format. Bridging with either Ethernet or 802.3 is uncomplicated. In bridging IPX SNAP format from Ethernet to FDDI or token ring and then back to Ethernet, SNAP packets are translated back into Ethernet format.

When translation bridging of the IPX Protocol involves an Ethernet backbone, the Novell file server MTU should be configured to be less than or equal to the MTU size of the Ethernet backbone (1,514 bytes).

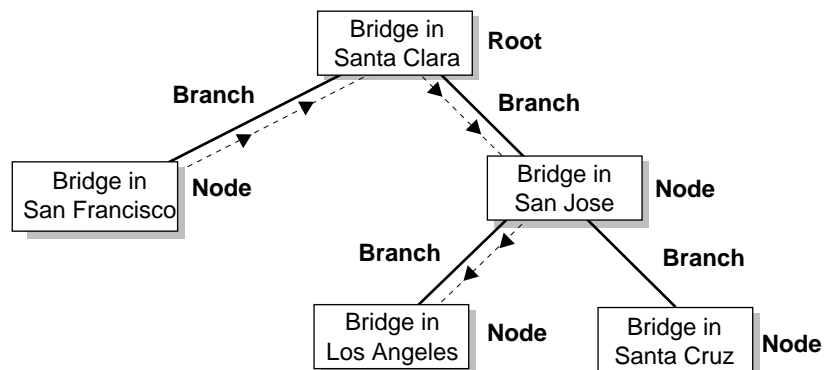
Spanning Tree Algorithm

The spanning tree algorithm detects loops and puts some bridge ports into blocking state, if necessary, so that only one route exists between any two stations. (A port in blocking state does not forward or receive packets.) Eliminating the extra paths creates a stable network configuration. When one or more bridges or ports in the stable topology fail, the algorithm automatically returns some ports from blocking state to forwarding state to ensure that all stations are connected.

For the spanning tree algorithm to be effective, all bridges in your extended network must run it.

An extended network without loops can be viewed as a spanning tree. A spanning tree is a topology in which one node is designated as the root, and any two nodes are connected to each other through one and only one route. Figure 27 is an example of the spanning tree structure in which one bridge represents the root, other bridges represent the nodes, and the communications lines represent the branches. The arrows illustrate the unique path that a packet from the San Francisco bridge takes when destined for the Los Angeles bridge. The topology would not be a spanning tree if there were also a line directly linking the San Francisco bridge and the San Jose bridge, or if the line between the San Jose bridge and the Santa Clara bridge were broken.

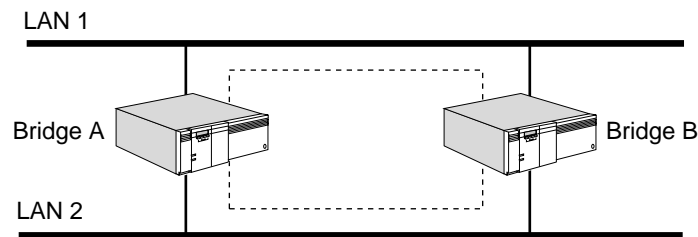
Figure 27 Spanning Tree Structure



When more than one bridge connects LANs, the network manager may inadvertently configure the network with loops, causing packets to circulate indefinitely. A loop exists if more than one path can be used to forward a packet from one end station to another. For example, the dotted line in Figure 28 highlights a loop; packets from a station on LAN 1 can be forwarded to one or more stations on LAN 2 via either bridge A or bridge B. The destination stations receive duplicate packets because both bridge A and bridge B replicate the packet and then forward the packet to LAN 2. If the station sends out a broadcast packet, both bridges forward it to their attached networks, creating packets that circulate indefinitely.

The spanning tree algorithm detects and breaks loops that can form within a bridging topology.

Figure 28 Network with a Loop



How the Algorithm Works

The spanning tree algorithm configures the network so that no loops exist in the extended network, and every two LANs can communicate with each other. This section lists the prerequisites required for the algorithm to work and gives an example to explain how the algorithm arrives at a loop-free configuration.

Algorithm Requirements for Configuring the Network

For the algorithm to configure the network:

- Each bridge must be able to recognize a unique destination address.
- Each bridge must have a unique identifier (bridge ID) that contains a priority field and a data link address.
- Each port of a bridge must have a unique identifier (port ID) that contains a priority field and a port number.
- Each port must be associated with a path cost, which is determined by the speed of its network interface (the faster the speed, the smaller the cost).

How the Algorithm Creates a Loop-free Configuration

To arrive at a loop-free configuration based on the bridge ID, port ID, and path costs, the algorithm performs the following tasks:

- Selects a bridge that acts as the root of the spanning tree network. This is usually the bridge with the lowest bridge ID of all the bridges on the extended network.
- Selects a root port on each bridge (except the root bridge) that incurs the lowest root path cost when the bridge forwards a packet to the root bridge.
- Selects the designated bridge on each LAN that incurs the lowest path cost when forwarding a packet from that LAN to the root bridge. The port through

which the designated bridge is attached to the LAN is called the designated port.

- Enables all root ports and designated ports so they can forward packets, and blocks all other ports.

The following example shows how the algorithm makes the selections, then eventually eliminates loops. Figure 29, Figure 30, Figure 31, and Figure 32 illustrate an extended network. In these figures, the bridges are numbered from 1 to 5, where bridge 1 has the lowest data link address, and bridge 5 has the highest.

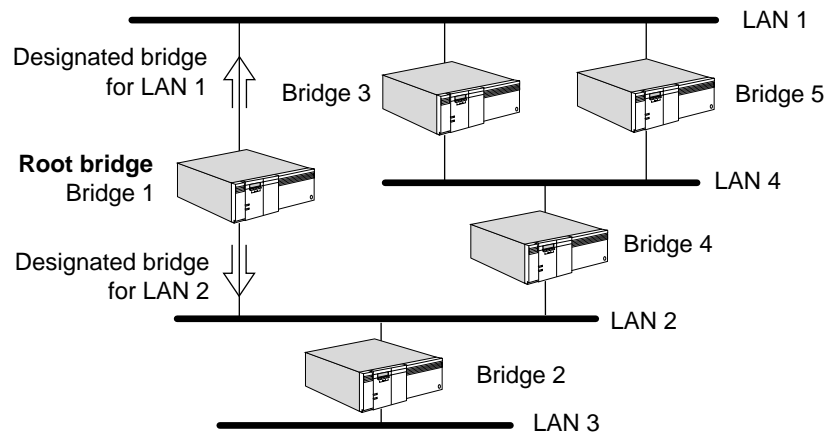
When the bridges are turned on, each assumes that it is the root bridge. Each bridge then transmits a packet called the Configuration Bridge Protocol Data Unit (CBPDU) through all its ports. A CBPDU contains information such as the ID of the bridge that the transmitting bridge considers the root bridge, the root path cost of the transmitting bridge, and the number of the source port.

When a bridge receives a CBPDU that contains superior information on one of its ports, it stores the information at that port. If this CBPDU is received at the root port of the bridge, the bridge also forwards it with an updated message to all attached LANs for which it is the designated bridge.

If a bridge receives a CBPDU on one of its ports that contains information inferior to that currently stored at that port, it discards it. If the bridge is a designated bridge for the LAN from which the CBPDU is received, it sends that LAN a CBPDU containing the up-to-date information stored at that port. In this way, inferior information is discarded and superior information is propagated on the extended network.

Assume that each port in Figure 29 is equipped with an Ethernet interface that has a path cost of 100, and that the priority fields in the IDs of bridge 1 and bridge 3 are the same. Having the lowest bridge ID (because its data link address is the lowest), bridge 1 becomes the root bridge, and its CBPDU is superior to the ones from other bridges. After exchanging a few CBPDUs and discarding the inferior ones, all bridges contain the same information that indicates that bridge 1 is the root bridge. Because a root bridge is automatically the designated bridge for all LANs to which it is attached, bridge 1 is also the designated bridge for LANs 1 and 2.

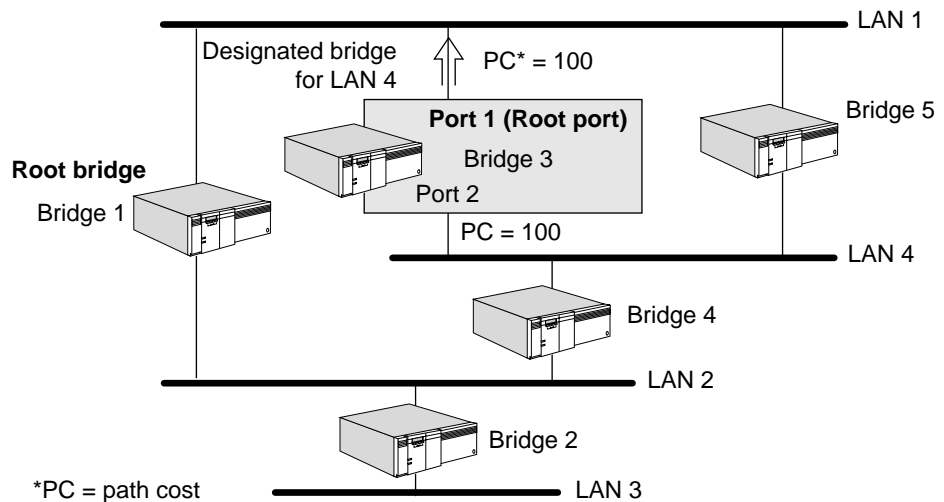
Figure 29 Root Bridge



Each bridge (except the root bridge) has to select a root port that will incur the least cost when the bridge forwards a packet to the root. The cost depends partly on the path cost of the port (determined by the speed of its network interface) and partly on the root path cost of the designated bridge for the LAN to which this port is attached.

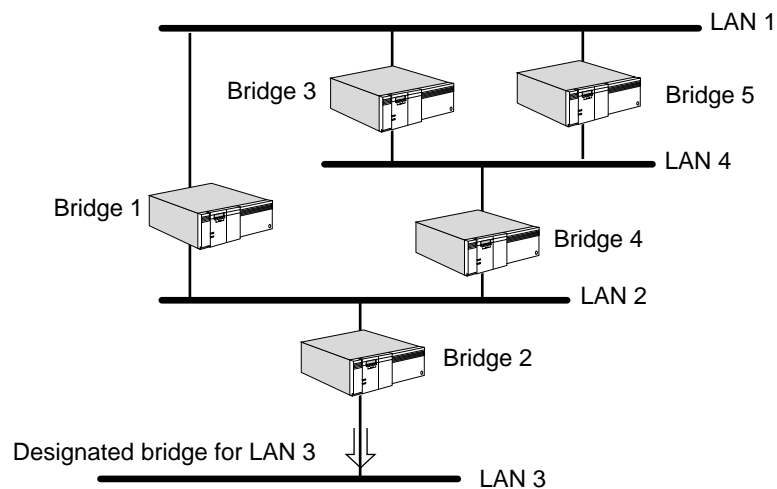
For example, in Figure 30, while ports 1 and 2 of bridge 3 both have the same network interface type and the same path cost, bridge 3 incurs less cost if it forwards a packet from port 1 than from port 2. The algorithm then decides that port 1 should be the root port for bridge 3.

Figure 30 Root Port



If a LAN is attached to a single bridge, that bridge is the designated bridge of the LAN. For example, in Figure 31, bridge 2 is the designated bridge for LAN 3, because bridge 2 is the only bridge attached to LAN 3.

Figure 31 Selecting a Designated Bridge when One Bridge Is Attached to a Network

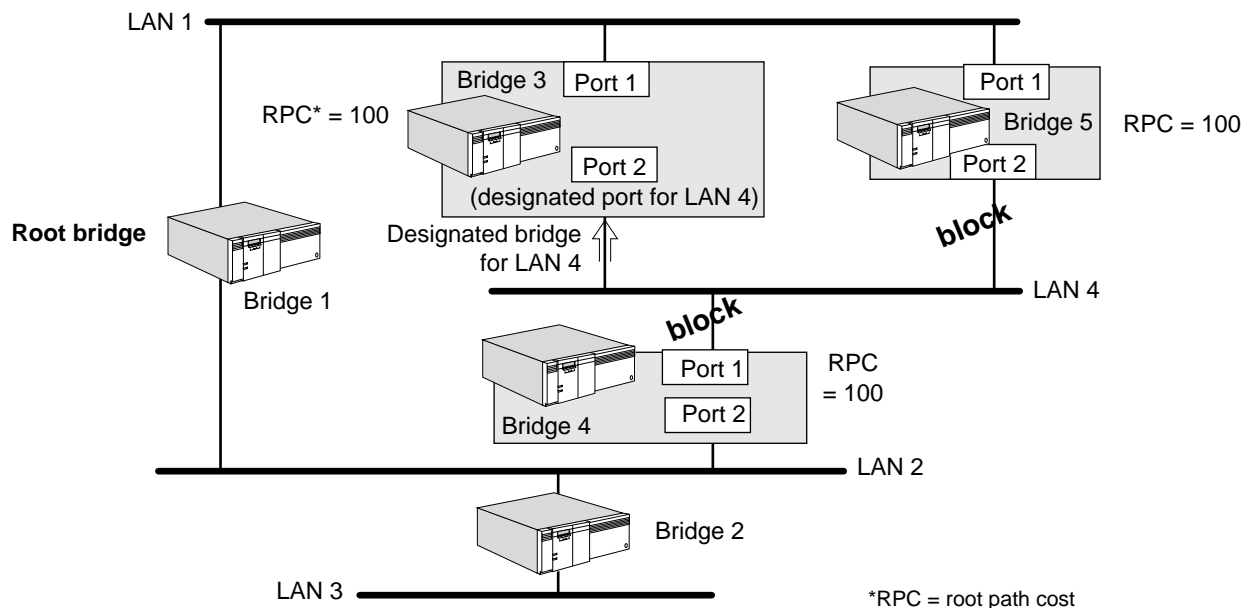


For a LAN that is attached to more than one bridge, a designated bridge must be selected. For example, in Figure 32, because LAN 4 is attached to bridge 3, bridge

4, and bridge 5, the algorithm must compare the root path costs of these bridges. In this case, their root path costs are the same. Having the lowest bridge ID, bridge 3 becomes the designated bridge for LAN 4. Because bridge 3 is attached to LAN 4 through port 2, port 2 is the designated port for LAN 4.

Bridge 1, which is the root bridge, is automatically the designated bridge for all attached LANs (that is, LANs 1 and 2). Because bridge 2 is the only bridge attached to LAN 3, it becomes the designated bridge for LAN 3.

Figure 32 Selecting a Designated Bridge when Multiple Bridges Are Attached to a Network



Only root ports and designated ports are put into forwarding state. Other ports, such as port 1 of bridge 4 and port 2 of bridge 5, are put into blocking state, as shown in Figure 32.

When a port is in forwarding state, it performs learning, filtering, and forwarding functions. When it is in blocking state, it performs none of these functions.

Because some ports are put into blocking state, none of the packets circulate on the extended network indefinitely.

Using the Algorithm with Wide Area Bridges

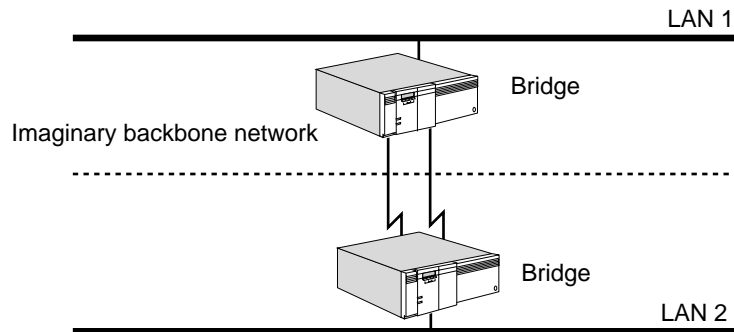
Although the examples in the previous section involve only local bridges, local and wide area bridges participate in configuring loop-free networks using the spanning tree algorithm.

In Figure 33, two bridges connect two remote networks. On each bridge, one of the interfaces is a network interface, and the others are serial links connected to the other wide area bridge. To apply the spanning tree algorithm in such a network configuration, it is assumed that the serial links are attached to an imaginary backbone network on which no end stations exist. The only traffic on the backbone is the traffic between the bridges. With this assumption, all bridge interfaces operate as if they were network interfaces, and the same spanning tree principle described above applies.

When wide area bridges with parallel lines, as shown in Figure 33, participate in the spanning tree algorithm, all remote links connected to the same wide area bridge are considered one network interface. The algorithm puts all links into either forwarding or blocking state. This ensures that the network topology can maximize the use of the bandwidth provided by parallel network links.

If you configure your wide area bridge with parallel lines, make sure that both paths are assigned to the same port. If you use separate ports, the spanning tree algorithm considers each port to be a separate network. As a result, one port will be put into blocking state. You can use parallel lines on different ports as a backup. If the line in the forwarding state fails, the second line moves from the blocking state to the forwarding state.

Figure 33 Two Wide Area Bridges Connected to Imaginary Backbone Network



Configuring the Spanning Tree Protocol over PPP

When you connect two bridges over a PPP serial link, both bridges must operate in the same spanning tree domain. 3Com supports the following configurations of the STP over PPP:

- Source route to source route
- Source route transparent to source route transparent
- Transparent bridge to transparent bridge
- Transparent bridge to source route transparent

The following configurations are not supported:

- Source route to transparent bridge
- Source route to source route transparent



If you connect bridges in the unsupported configurations, the separate spanning tree domains are combined into a single domain.

When two bridges are connected over a PPP serial link, both bridges must be operating in the same spanning tree domain (SR or TB/SRT). The following configurations are supported:

- SR-SR
- SRT-SRT
- TB-TB
- TB-SRT

The following configurations are not supported:

- SR–TB
- SR–SRT

If you connect bridges in the unsupported configurations, the separate SR and SRT/TB spanning tree domains will combine into a single spanning tree domain.

A bridge is configured for SRT, SR, or TB modes as follows:

- SRT
One or more ports are configured for transparent bridging and one or more ports are configured for source route bridging.
- SR
One or more ports are configured for source route bridging and no ports are configured for transparent bridging.
- TB
One or more ports are configured for transparent bridging and no ports are configured for source route bridging.

Configure ports for transparent bridging by setting the `TransparentBRidge` parameter in the `BRIge` Service. Configure ports for source route bridging by setting the `SrcRouBridge` parameter in the `SR` Service.

Spanning Tree Addressing

Transparent and source route transparent bridges participate in a spanning tree domain, which is identified when the destination address field of the spanning tree packet is the hexadecimal group address 0180C2000000. Source route bridges participate in a different spanning tree domain, which is identified when the destination address field of the spanning tree packet is the hexadecimal bridge functional address 030000008000. Both addresses are shown in canonical addressing format.

If a bridge has different types of bridging enabled on different ports, the spanning tree algorithm determines what type of bridge it is overall (transparent, source route, or source route transparent) according to the following criteria:

- If a bridge does not have transparent bridging enabled on any ports and has source route bridging enabled on at least one port, it is considered a source route bridge.
- If a bridge has transparent bridging enabled on at least one port and source route bridging enabled on at least one port, it is considered a source route transparent bridge.
- If a bridge does not have source route bridging enabled on any ports, it is considered a transparent bridge.

The spanning tree algorithm detects loops independent of the operating mode of the bridge.

Modifying Spanning Tree Parameters

The Spanning Tree Protocol (STP) Service controls parameters used by the spanning tree algorithm (for example, the priority field in the bridge identifier) to influence

the final network configuration. For more information on setting STP parameters, see the STP Service Parameters chapter in *Reference for Enterprise OS Software*.

Reconfiguring the Topology

The spanning tree algorithm reconfigures the network topology when bridges are added or removed, or when the network manager changes the parameters.

Whenever a bridge detects a topology change, if it is a designated bridge for a LAN, it sends out a topology change notification Bridge Protocol Data Unit (BPDU) through its root port. This information is eventually relayed to the root bridge. The root bridge then sets the topology change flag in its CBPDU so that the information is broadcast to all bridges. It transmits this CBPDU for a fixed amount of time to ensure that all bridges are informed of the topology change.

If a port is changed from blocking state to forwarding state as a result of the topology change, the algorithm ensures that it propagates the topology information to all ports before that port starts forwarding data. This prevents temporary data loops.

If a bridge does not receive packets from an address within a fixed period of time, it removes that address from its routing table. After reconfiguration, the bridge removes these addresses faster to ensure that each active port still forwards packets to the right network after a topology change.

Load Sharing

When multiple paths are assigned to a port on a NETBuilder II bridge/router, a load-sharing algorithm is used. The load-sharing algorithm selects the highest bandwidth line as the primary line. Any outgoing data is transmitted through this line until a certain threshold (defined within software limits for that bandwidth) is reached. When the threshold is reached, packets are forwarded on the next highest bandwidth line. If the number of bytes queued on the primary line falls below the threshold, outgoing packets revert to the primary line.

Routing Tables

A bridge forwards packets according to information in the routing table. Each entry in this table lists an address, the network on which the station with that address can be found, and an indication of elapsed time since a packet was received from that node. For an interpretation of the routing table, see the BRidge Service Parameters chapter in *Reference for Enterprise OS Software*.

The two types of routing table entries are: learned (dynamic) entries and user-assigned (static or permanent) entries.

- Learned entries are entries that the bridge learns from the network. The learned entries are subject to dynamic changes or deletion whenever the -BRidge CONTrol parameter is set to Aging and LEarn.
- User-assigned entries are entries assigned by entering ADD -BRidge ROute. The user-assigned entries can be changed or deleted manually only through the ADD or DElete commands.

You can access the routing table of transparent bridges by entering:

```
SHow -BRidge AllRoutes
```

For complete information on this parameter, see the BRidge Service Parameters chapter in *Reference for Enterprise OS Software*.

You can configure the size of the routing table on the transparent bridge using the -BRidge RouteTableSize parameter. For complete information on this parameter,

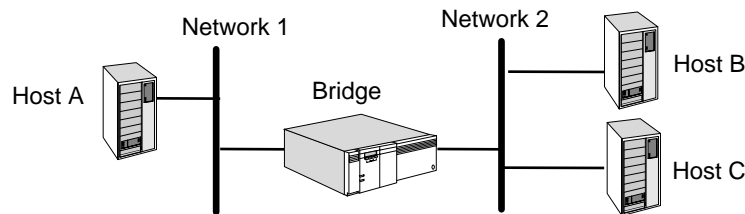
see the BRidge Service Parameters chapter in *Reference for Enterprise OS Software*.

Learning and Filtering

This section describes how a bridge learns the network configuration and adapts to the addition or removal of stations on the attached network segment in order to perform standard filtering. For information on 3Com mnemonic filtering and related filtering processes such as logging, sequencing, and packet prioritization, see the Configuring Mnemonic Filtering chapter. For complete explanations of packet filtering parameters, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Figure 34 shows two networks interconnected by a bridge. After the bridge receives a packet, it decides whether to forward it to the other network or discard it. To help make this decision, the bridge determines to which network the destination of the packet belongs.

Figure 34 Bridge Learning



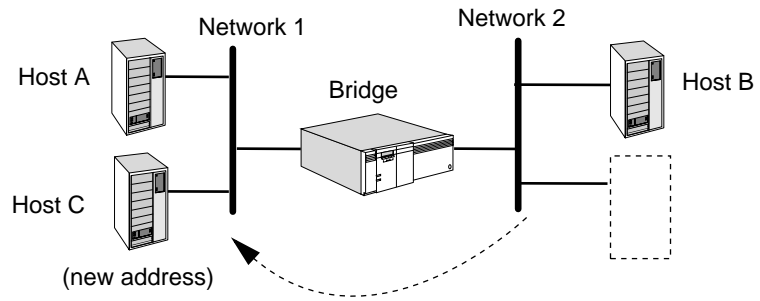
When a bridge is operating, it receives packets from all attached networks. By looking at the source address of packets, the bridge learns the addresses of stations on each network and stores them in its routing table. For example, when the bridge in Figure 34 receives a packet from network 1 with the address for host A as the source address, it learns that host A is on network 1. In the same way, it also learns that hosts B and C are on network 2.

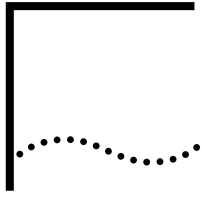
If a packet is destined for the network where it originated, the bridge discards it. This is called standard filtering. For example, if the bridge in Figure 34 receives a packet on network 2 from host C that is addressed to host B, and it determines from the learned entries in its routing table that host B is on the same network as host C, then it discards the packet.

In addition, the bridge uses the learned network configurations to forward packets destined for another network. For example, if the bridge receives a packet on network 2 from host C addressed to host A, it determines that host A is on another attached network, and forwards the packet to that network.

If the bridge receives a packet from a host on a network that has not yet been learned, the bridge forwards the packet to all ports except the port on which the packet was received.

The bridge also can learn that a station has been removed from one of its attached networks. For example, in Figure 35, host C was moved from network 2 to network 1. The bridge no longer receives packets on network 2 with host C as the source address. The bridge record of the location of host C is no longer updated and is removed (aged) from the routing table. With host C attached to network 1, the bridge receives packets from network 1 with the address of host C as the source address, and learns that network 1 now includes host C.

Figure 35 Network Configuration after Host C Is Moved



CONFIGURING SOURCE ROUTE BRIDGING

This chapter describes the minimum steps you must perform to configure your source route bridge and various ways to customize the configuration. It also describes how to troubleshoot the source route bridge and provides basic information on how it works.



For conceptual information, see “How the Source Route Bridge Works” later in this chapter.

Configuring a Basic Source Route Bridge

This section describes how to configure a source route bridge to operate in a token ring or Fiber Distributed Data Interface (FDDI) environment. (NETBuilder II bridge/routers only support an FDDI environment.) In this section, a network with multiple rings or other network segments is called an *extended network*. For information on how to configure a source route bridge to operate in a wide area networking environment, see “Configure Source Route Bridging over a Wide Area Network” later in this chapter.



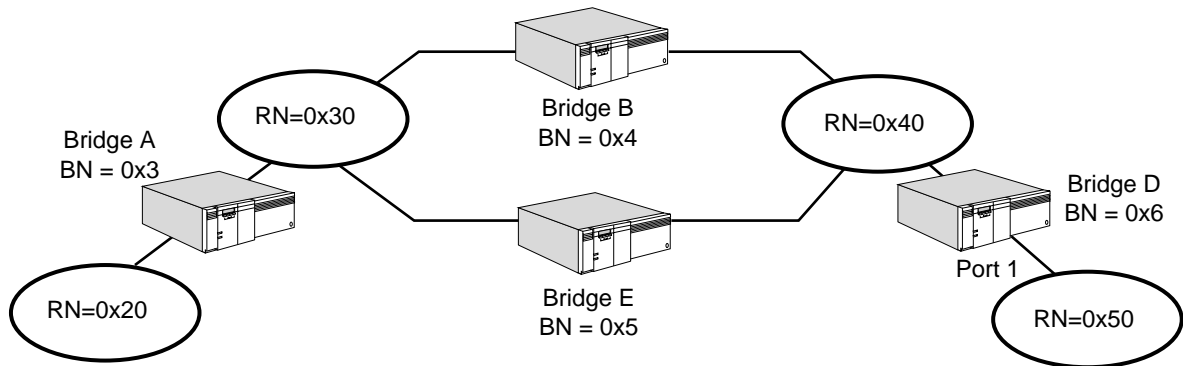
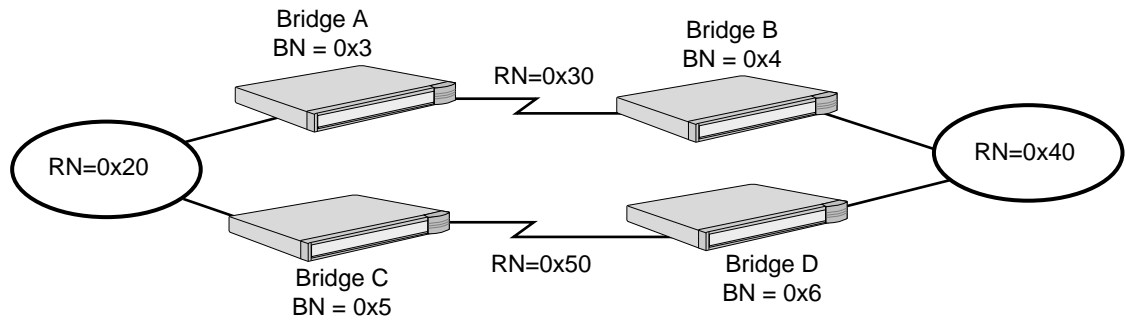
Source route bridging is supported on token ring, FDDI, Point-to-Point Protocol (PPP), Frame Relay, Asynchronous Transfer Mode (ATM), Asynchronous Transfer Mode data exchange interface (ATM DXI), X.25, Switched Multimegabit Data Service (SMDS), and Integrated Services Digital Network (ISDN). Also, configuring source route bridging can affect IBM-related services such as SDLC or DLSw. For more information, see “Configuring LLC2 with Other Services” in the Configuring the LLC2 Data Link Interface chapter.

Prerequisites

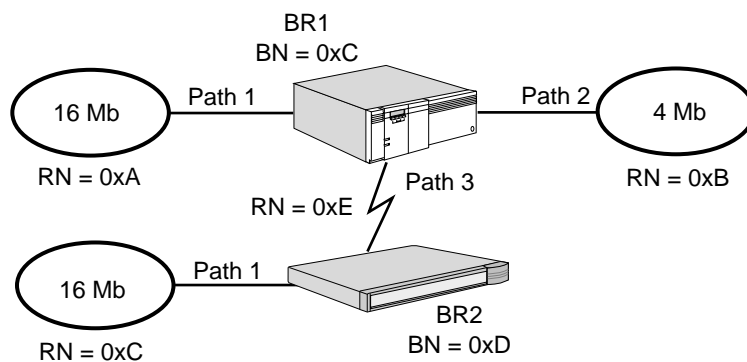
This section assumes that you have logged on to the system with Network Manager privilege and set up the ports and paths of your source route bridge according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Before setting up a source route bridge, you need to examine your network topology and generate the following:

- A unique number for each ring in an extended network. For example, in the topology shown in Figure 36, the four rings have been assigned hexadecimal ring numbers (RNs) 0x20, 0x30, 0x40, and 0x50. In Figure 37, the two rings have been assigned hexadecimal RNs 0x20 and 0x40. In addition, the serial interfaces have been assigned the hexadecimal RNs 0x30 and 0x50.
- A unique number for each bridge in a set of parallel bridges. For example, in the topology shown in Figure 36, parallel bridges B and E have been assigned the hexadecimal bridge numbers (BNs) 0x4 and 0x5. (When more than one bridge interconnects the same networks, the bridges are called *parallel bridges*.)

Figure 36 Sample FDDI or Token Ring Topology Using NETBuilder II Bridge/Router**Figure 37** Sample Token Ring Topology Using SuperStack II Bridge/Routers

Procedure Figure 38 shows a sample token ring topology, which you can see while performing this procedure.

Figure 38 Source Route Bridging Sample Topology

To configure a source route bridge, follow these steps:

- 1 If you are configuring a source route bridge to operate in an FDDI environment, skip this step and go to step 3. If you are configuring a source route bridge to operate in a token ring environment, you may need to set the ring speed of each path.

The default ring speed is 4 Mb. If your source route bridge is a NETBuilder II bridge/router, you need to perform this step only if your network is composed of 16 Mb rings. If your source route bridge is a model 32x or 52x SuperStack II

NETBuilder bridge/router, the ring speed is automatically detected upon startup. You need to perform this step only if your bridge is connected to an intelligent hub and your network is composed of 16 Mb rings.

For example, to set the ring speed of path 1 of BR1 and BR2 (as shown in Figure 38) to 16 Mb, enter the following command on both bridges:

```
SETDefault !1 -PATH BAud = 16000
```

A message similar to the following appears:

Note: You must Enable `-PATH CONTROL` for this Path parameter to take effect.

- 2 Enable the paths you set the ring speed for in step 1 using:

```
SETDefault !<path> -PATH CONTROL = Enabled
```

A message similar to the following appears:

```
Thu Jan 1 09:09:14 1995 Path 1 available
```

At this point, connect the DB9 end of the token ring cable that leads from the 16 Mb ring to the token ring interface on your bridge/router.

It will take a minute or two for path 1 to start operating. When path 1 is operational, the system responds with a display similar to the following:

```
Thu Jan 1 09:12:36 1995 Path 1 UP
```

- 3 Assign each bridge port on your network the ring number of the network it accesses.

If you are setting up a pure router to forward packets to end systems on an extended network, skip this step.

To assign a ring number, use:

```
SETDefault !<port> -SR RingNumber = <number> (1-4095) | 0x<number> (1-FFF]
```

For example, to assign the hexadecimal ring number 0xA to BR1 path 1, as shown in Figure 38, enter:

```
SETDefault !1 -SR RingNumber = 0xA
```

To assign the equivalent decimal ring number to BR1 path 1, enter:

```
SETDefault !1 -SR RingNumber = 10
```

A serial line running PPP, Frame Relay, ATM DXI, SMDS, or X.25 is treated as a virtual ring.

- 4 Assign a different bridge number to each bridge in a set of parallel bridges using:

```
SETDefault !<port> -SR BridgeNumber = <number> (0-15) | 0x<number> (0-F)
```

If your network is not composed of parallel bridges, you do not need to assign a unique bridge number to each bridge. You can use the default setting of 3.

To assign the hexadecimal bridge number 0xC to a bridge, enter:

```
SETDefault -SR BridgeNumber = 0xC
```

To assign the equivalent decimal bridge number to a bridge, enter:

```
SETDefault -SR BridgeNumber = 12
```

5 Enable global bridging on each bridge.

For example, enable bridging on BR1 and BR2 by entering the following command on each bridge:

```
SETDefault -BRidge CONTrol = Bridge
```

Source route bridging is enabled by default on all ports (the default setting of the -SR SrcRouBridge parameter is SrcRouBridge) and source route bridging should begin to operate after you assign a ring number and enable global bridging.

6 If you do not want to operate in source route transparent (SRT) mode, disable per-port transparent bridging using:

```
SETDefault !<port> -BRidge TransparentBridge = NoTransparentBRidge
```

Transparent bridging is not supported on models 32x and 52x SuperStack II NETBuilder bridge/routers. You do not need to perform this step for this model.

After you complete this procedure, go to “Verifying the Configuration” later in this chapter.

Configure Source Route Bridging over a Wide Area Network

You can configure your source route bridge to forward packets over the following types of wide area networks:

- PPP
- Frame Relay
- Asynchronous Transfer Mode (ATM)
- Asynchronous Transfer Mode Data Exchange Interface (ATM DXI)
- X.25
- SMDS

Source Route Bridging over PPP

For complete information on configuring PPP, see the Configuring Wide Area Networking Using PPP chapter.

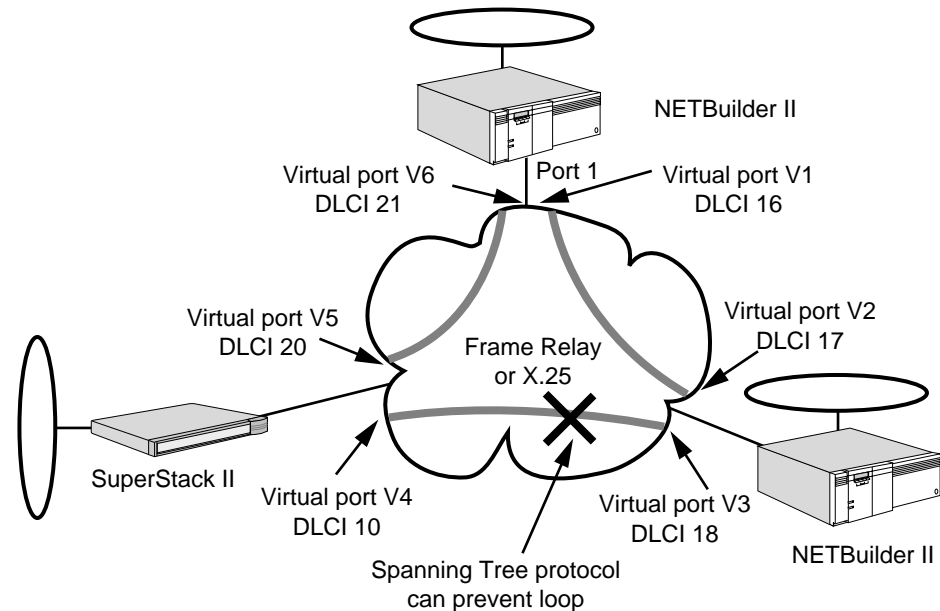
Source Route Bridging over Frame Relay, ATM, ATM DXI, and X.25

Source route bridging over Frame Relay, ATM, ATM DXI, and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to source route bridge over a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. For complete information on configuring source route bridging over Frame Relay, ATM, or ATM DXI, including a discussion of fully meshed, partially meshed, or nonmeshed topologies and virtual ports, see the Configuring Wide Area Networking Using Frame Relay chapter, the Configuring Internetworking Using ATM chapter, and the Configuring Wide Area Networking Using the ATM DXI chapter. For complete information on configuring source route bridging over X.25, including a discussion of fully meshed, partially meshed, or nonmeshed topologies and virtual ports, see the Configuring Wide Area Networking Using X.25 chapter. For information on the number of virtual ports supported per platform, see Table 11 in the Configuring Advanced Ports and Paths chapter.

When creating virtual ports over a heavily trafficked partially meshed or nonmeshed topology, 3Com recommends that each source route bridge on the Frame Relay, ATM, ATM DXI, or X.25 network have a permanent virtual circuit for the proper operation of the Spanning Tree Protocol. Figure 39 shows a network

composed of two NETBuilder II bridges and a model 327 SuperStack II bridge connected by virtual ports. The interconnection of the three source route bridges causes a potential loop. The Spanning Tree Protocol can prevent this loop by blocking a route as shown in Figure 39.

Figure 39 Source Route Bridging Over Frame Relay or X.25 in a Nonmeshed Topology with a Potential Loop



Source Route Bridging over SMDS

Source route bridging over SMDS is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure source route bridging over SMDS, see the Configuring Wide Area Networking Using SMDS chapter.

Source Route Bridging over ISDN

For information on wide area networking using Integrated Services Digital Network (ISDN), see the Configuring Wide Area Networking Using ISDN chapter.

Verifying the Configuration

After you configure your source route bridge, you need to verify its configuration by following these steps:

- 1 Check the state of the current configuration by entering:

```
SHoW -SR DIAGNostics
```

The display provides troubleshooting information about source route configuration errors and gives suggestions for corrective actions.

- 2 Check the configuration of the source route bridge and the status of each port and path by entering:

```
SHoW -SR CONFiguration
```

The display shows the source route bridging status on a per-port basis. A Forwarding status indicates that source route bridging is activated. A Down status indicates that source route bridging is not activated because of one or a combination of the following conditions:

- The port or path is disabled.
- The -BRidge CONTrol parameter is set to NoBridge.
- The -SR SrcRouBridge parameter is set to NoSrcRouBridge.
- A ring number has not been assigned to the port.

The display also indicates if end system source routing or source route transparent bridging gateway (SRTG) is enabled. For more information, see “Guidelines for Per-Port Route Discovery” and “Configuring Source Route Transparent Bridging Gateway” later in this chapter.

3 If the display indicates that a port or a path is down, follow these steps:

- a** Check the configuration of each port by entering:

SHoW -PORT CONFIguration

- b** Check the configuration of each path by entering:

SHoW -PATH CONFIguration

4 Test the source route bridge by sending packets across it.

For example, make a connection from a device on one attached network to a host on another attached network. If you can successfully make a connection, the source route bridge is ready for normal operation; otherwise, see “Troubleshooting the Configuration” later in this chapter.

Getting Statistics

After your source route bridge is up and running, you may want to gather statistics. For information on interpreting the statistics display, see the Statistics Displays appendix.

You can collect statistics for a specific time period by using the -SYS SampleTime and -SYS STATistics parameters. For more information, see *Reference for Enterprise OS Software*.

To gather statistics, follow these steps:

1 Display source route bridging statistics for all ports by entering:

SHoW -SYS STATistics -SR

2 Display statistics for all ports by entering:

SHoW -SYS STATistics -PORT

3 Check the statistics for all paths by entering:

SHoW -SYS STATistics -PATH

If the display indicates that there are errors (for example, cyclic redundancy check errors) on the attached network, check:

- That the transceiver cable is properly attached to the transceiver.
- That the transceiver is properly attached to the network cable.
- That the network is properly terminated.

If the errors happen on a serial line, check:

- Cable attachments.
- Channel service unit/digital service units (DSU/CSUs).
- Modems on each end of the serial line.

If the line is a leased line, request help from the company that leases the line (for example, the telephone company).

Troubleshooting the Configuration

To troubleshoot the source route bridge, follow these steps:

- 1 Check for configuration errors using:

```
SHoW [!<port>] -SR DIAGnoStics
```

The display provides troubleshooting information about source route configuration errors and gives suggestions for corrective actions.

- 2 Access source route bridge configuration information and check the status of each path. Verify that each path is assigned to the appropriate network by entering:

```
SHoW -SR CONFIguration
```

Make sure that the status of the source route bridge is Forwarding. Verify that the path is enabled by entering:

```
SHoWDefault -PORT CONFIguration
```

```
SHoWDefault -PATH CONFIguration.
```

- 3 Display all learned remote routes using:

```
SHoW [!<port>] -SR WanRoutes
```

SHoW displays all the currently learned source routes and the associated DLCI, SMDS individual address, or X.25 DTE address for each learned route. If the port is specified, the display for port-related parameter values is limited to that port.

- 4 If the display in step 1 indicates that a port or path is down, follow these steps:

- a Check the configuration of each port by entering:

```
SHoW -PORT CONFIguration
```

- b Check the configuration of each path by entering:

```
SHoW -PATH CONFIguration
```

- 5 Check for other activity on the source route bridge through the statistics display.

- a For a detailed accounting of errors on a given port, enter:

```
SHoW -SYS STATistics -SR
```

If there is no other activity on the source route bridge, check its physical attachments to other networks, including boards, back panel connectors, and transceiver or modem connectors. For lines to wide area bridges, check the DSU/CSU or modem and its configuration.

- b If a large number of errors occur on a bridge's local or serial line to a network, check the physical lines.

For a detailed accounting of errors on a given path, enter:

```
SHoW -SYS STATistics -PATH
```

Some statistics can be set to zero using the FLush -SYS STATistics command to provide a starting point for subsequent analysis of these reports.

- 6 If possible, replace any bridge you suspect has problems with another bridge or a repeater. Check to see if the problem persists.

If the problem persists, then the bridge is not the cause of the problem.

To determine whether a pair of source route bridges can communicate with each other, use the data link test. This test allows the bridges to exchange test packets

and display the related statistics. Use the DLTest command, which is described in the Commands chapter in *Reference for Enterprise OS Software*.

Related Information

End systems on token ring report soft errors such as frame-copied errors through the media access control (MAC) Report Error frame. End systems may generate a small number of MAC Frame Copy error report packets when a NETBuilder II Bridge is initializing. For the NETBuilder II system to learn addresses on the token ring, it copies the packet with the unknown source address and sets the address-recognized (A) and frame-copied (C) bits in the Frame Status (FS) field (1 byte) located at the end of the frame after the Frame Check sequence and the Ending Delimiter field.

A problem occurs when the FS (A) and (C) bits have been set and the destination of the frame is a local end system. The end system normally sets the (A) and (C) bits, and when it receives a frame with these values already set, it reports an error. These errors are counted until the error threshold is reached; then a MAC Report Error is sent out by the end system.

Customizing the Source Route Bridge

Table 9 summarizes the features that allow you to customize your source route bridge and which platforms each feature is supported on.

Table 9 Source Route Bridge Features/Platforms Supported

Source Route Bridge Feature	NETBuilder II	Model 32x and 52x SuperStack II NETBuilder
Per-port source route bridging	Yes	Yes
Per-port source route transparent bridging	Yes	No
Source route transparent bridging gateway (SRTG)	Yes	No
IBM connectivity	Yes	Yes
Largest frame size	Yes	Yes
Passive bridging	Yes	No
Spanning tree in a source route bridging environment	Yes	Yes
Parallel bridges	Yes	No
Broadcast traffic reduction	Yes	Yes
Explorer frame propagation	Yes	Yes
Filters	Yes	Yes
Security	Yes	Yes
Configuration as an end system	Yes	Yes
Per-port route discovery for end system source routing		
Utility for discovering routes to an end system		
Static routes		
Aging entries in the routing table		
Token access priority		

This section briefly describes and explains how to set up the source route bridging features. Not all available parameters are discussed in this section. For more

information on all available parameters, see the SR Service Parameters chapter in *Reference for Enterprise OS Software*.

Enabling and Disabling Per-Port Source Route Bridging

By default, source route bridging is enabled on all ports. You can disable source route bridging on specified ports using:

```
SETDefault !<port> -SR SrcRouBridge = NoSrcRouBridge
```

To enable source route bridging, use:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```



*For source route bridging to take effect on a port, the port must additionally be enabled (as described in the *Configuring Basic Ports and Paths* chapter), the -BRidge CONTrol parameter must be set to Bridge (as described in the *Configuring Bridging* chapter), and ring numbers must be assigned (as described in "Configuring a Basic Source Route Bridge" earlier in this chapter).*

For complete information on the -SR SrcRouBridge parameter, see the SR Service Parameters chapter in *Reference for Enterprise OS Software*.

Enabling and Disabling Per-Port Source Route Transparent Bridging

This feature is not supported on model 32x and 52x SuperStack II NETBuilder bridge/routers.

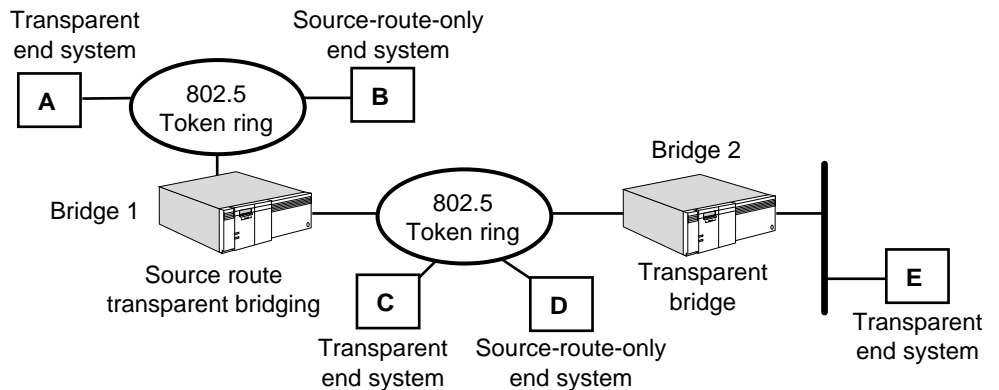
If your token ring or FDDI network is composed of users on transparent (non-source route) end systems as well as source route end systems as shown in Figure 40, you can enable transparent bridging on your source route bridge. By enabling source route transparent bridging, your source route bridge can forward source route or transparent bridged frames. For conceptual information, see "Source Route Bridging" later in this chapter.

By default, source route transparent bridging is enabled on all ports. As shown in Figure 40, Bridge 1 has source route transparent bridging enabled, which allows the transparent end systems A, C, and E to communicate. The source route end system B can communicate with the source-route-only end system D. However, the source-route-only end systems B and D cannot communicate with transparent only end systems A, C, or E.

If you want to disable transparent bridging on some ports, use:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

Figure 40 Source Route Transparent Bridging



Configuring Source Route Transparent Bridging Gateway

This feature is not supported on model 32x and 52x SuperStack II NETBuilder bridge/routers.

You can connect source route and transparent bridging domains, and allow communication between the two by configuring SRTG.

The SRTG feature is supported on the NETBuilder II platform and on all LAN and WAN media currently offered by 3Com.

The SRTG bridges only logical link control, type 2 (LLC2) and NetBIOS traffic between source route and transparent bridging domains. SRTG supports both 802.3 and Ethernet Version II frames on Ethernet, and supports multiple paths between source route and transparent bridging domains (only one path is active at a time because the SRTG detects and breaks loops according to the spanning tree algorithm).

For conceptual information about SRTG, see "Source Route Transparent Bridging Gateway Concepts" later in this chapter.

Prerequisites

This section assumes that you have logged on to the system with Network Manager privilege and set up the ports and paths of your source route bridge.

Procedure

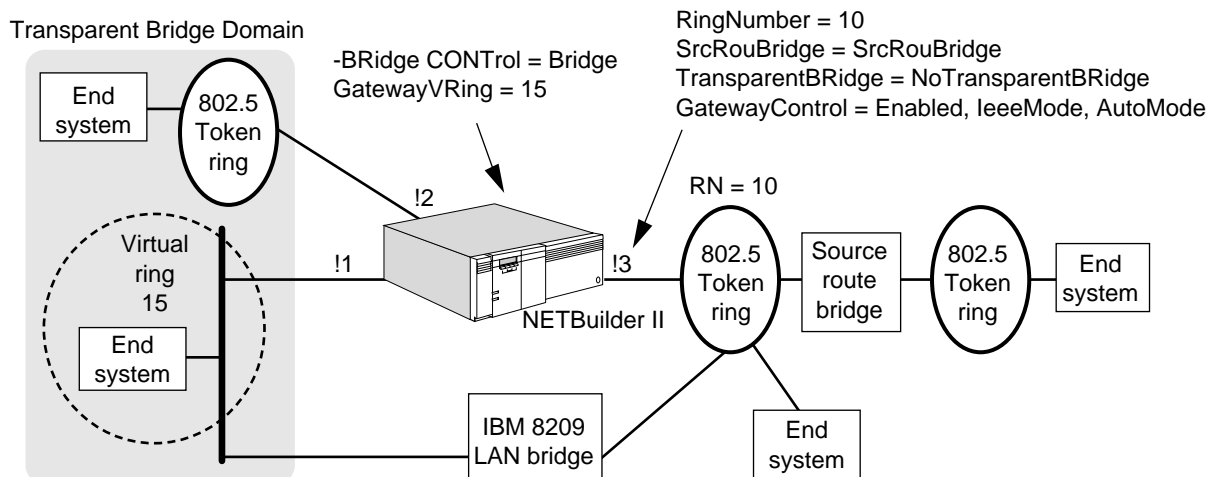
To configure the SRTG to support bridging of LLC2 and NetBIOS traffic between source route and transparent bridging domains, see Figure 41 and follow these steps:



You cannot perform both transparent bridging and source route bridging on a port being used for the SRTG. You can perform either transparent or source route bridging, but not both at the same time.

- 1 Configure the basic source route bridge on the NETBuilder II source route port connected to the source route domain by referring to "Configuring a Basic Source Route Bridge" earlier in this chapter.

Figure 41 Source Route Transparent Bridging Gateway Configuration



- 2 Verify that source route bridging is enabled on the source route port by entering:

```
Show -SR SrcRouBridge
```


If it is disabled, enable it using:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```

- 3 If no transparent bridging stations exist in the source route domain, disable transparent bridging on the source route port using:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

- 4 Verify that transparent bridging is enabled on the transparent bridging port using:

```
SHow -BRidge TransparentBRidge
```

If transparent bridging is not enabled on the specified port, use:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

- 5 Configure a virtual ring number for the transparent bridging domain using:

```
SETDefault -SR GatewayVRing = <number>(1-4095) | 0x<number>(1-FFF)
```

You can enter the virtual ring number in decimal or hexadecimal (precede the hexadecimal number with a 0x as indicated in the syntax).

Before forwarding packets from the transparent bridging domain, SRTG adds the virtual ring number and its own bridge number to the source route information of the destination station retrieved from the source route table. From point of view of a source route station, the entire transparent bridge LAN appears as a single source route ring.

- 6 Enable SRTG on both the source route and transparent bridging ports, and set the encapsulation format on the transparent bridging port (Ethernet only) by using:

```
SETDefault !<port> -SR GatewayControl = ([Enabled | Disabled], [IeeeMode | EtherMode], [AutoMode | NoAutoMode])
```

Select Enabled to enable SRTG.

The combination of the next two pairs of settings determines what encapsulation format is used when translating token-ring LLC-based packets.

If NoAutoMode is selected, SRTG does not keep track of the encapsulation format of each transparent bridging station. The final encapsulation method is determined by EtherMode (Ethernet II encapsulation with packet type of 0x80D5) or IeeeMode (IEEE 802.3 encapsulation) settings when the packets are bridged to the Ethernet domain.

Select AutoMode if you want SRTG to automatically keep track of the encapsulation format of each station. If AutoMode is selected, different packet translation rules are used for known stations and unknown stations. For known stations, the IeeeMode | EtherMode settings are ignored and the encapsulation format learned for those stations is used. For unknown stations, LLC-based packets are translated into both 802.3 and Ethernet Version II frames.



The DSAP field in the token ring 802.2 frame must be a multiple of 4s (00, 04, 08, and so forth) except BC and E0, which are reserved for Banyan VINES and IPX, respectively.

For more information about frame conversion, see "Frame and Address Conversion" later in this chapter.

After SRTG is enabled, packets are bridged between the source route and transparent bridge domains.



Do not enable both data link switching (DLSw) and SRTG on the same port because packet duplication may occur if both features connect the same areas.

Related Information

If your SRTG topology includes a transparent bridge in the transparent bridge domain and your application involves NetBIOS and Systems Network Architecture (SNA) traffic that uses functional addresses as a destination address, you may have to add a mapping between the functional and multicast address on the transparent bridge if the destination and source media types are different. Use the `-BRidge FunctionalAddr` parameter. For more information, see “Translation Bridging” and “Adding Functional-Address-to-Multicast- Address Mappings to the Default Table” in the Configuring Bridging chapter.

Connecting IBM Bridges to 3Com Token Ring Bridges

This section provides information on connecting 3Com token ring bridges to IBM bridges.

Procedure

For complete information on setting your 3Com token ring bridge to source route or source route transparent mode, see “Configuring a Basic Source Route Bridge” and “Enabling and Disabling Per-Port Source Route Transparent Bridging” earlier in this chapter.

Related Information

Some IBM bridges support source route-only mode. When configuring these bridges and 3Com token ring bridges in the same network environment, you must configure the 3Com bridge in either source route or source route transparent mode. For more information about source route and source route transparent mode, see “Source Route Bridging” and “Source Route Transparent Bridging” later in this chapter.

IBM bridges support the hexadecimal-only format for bridge and ring numbers. The 3Com token ring bridge supports entry of both decimal and hexadecimal format for the `-SR RingNumber` and `-SR BridgeNumber` parameters. A hexadecimal format entry must be preceded by a `0x`, as shown in the following examples:

```
SETDefault !1 -SR RingNumber = 0xA
```

The ring number is displayed as decimal 10 with the hexadecimal equivalent in parentheses.

```
SETDefault -SR BridgeNumber= 0xF
```

The bridge number is displayed as decimal 15 with the hexadecimal equivalent in parentheses.

The IBM PC LAN Bridge is not fully compatible with the 3Com token ring implementation of Spanning Tree Protocol in a parallel bridge configuration. In this configuration, the 3Com token ring bridge forwards single-route broadcast frames. When configuring the IBM PC LAN Bridge in a parallel bridge configuration with a 3Com token ring bridge, set the 3Com bridge as source route-only mode. The IBM PC LAN Bridge sends out a broadcast test packet before it can become fully operational to ensure that IBM bridge adapters are not on the same ring. A parallel 3Com token ring bridge in source route transparent or

transparent mode can forward this test packet, confusing the IBM PC LAN Bridge and preventing it from coming up. To ensure that the two parallel bridges come up, the 3Com token ring bridge must be in source route-only mode.

Configuring the Largest Frame Size

The LargestFrameSize parameter specifies the maximum size frame that can be sent and received on a port. The source route bridge negotiates the largest frame size of all transit routes down to this size.

Use this parameter to regulate the amount of data transmitted by end systems to prevent time-outs due to slow network links. If the connected network contains low-speed WAN links, assign a lower largest frame size value.

The base values specified in IEEE 802.1D are supported and are listed in Table 10. Extended values listed in the IEEE specification are not currently supported.

Table 10 Valid Largest Frame Size Values

LargestFrameSize Parameter Setting	Data Unit Length
0	516 octets
1	1,470 octets
2	2,052 octets
3	4,399 octets
4*	8,130 octets
5*	11,407 octets
6*	17,749 octets
7*	41,600 octets

* These values are not supported.

By default, 3Com bridge/routers use a setting of 3, which is equivalent to a frame size of 4,399 octets.

The value can be changed using:

```
SETDefault !<port> -SR LargestFrameSize = <number>(0-7)
```



The maximum physical frame size that can be received and forwarded by a NETBuilder II system with a Token Ring or Token Ring + module and model 32x and 52x SuperStack II bridge/routers is 4,500 bytes.

Configuring Passive Bridging

This feature is not supported on model 32x and 52x SuperStack II bridge/routers.

To work around the bridge/router hop-count limitation for token ring networks consisting of eight or more rings, you can configure the attached source route bridges for passive bridging and effectively create one logical ring from multiple rings. Creating logical rings allows you to work around the token ring adapter limitation on the maximum number of rings in the route designator fields.

Procedure

To configure passive bridging on a network similar to Figure 42, follow these steps:

- 1 Configure the bridges that are within the logical ring.
 - a Enable passive bridging.

For example, on bridge 1 and bridge 2, enter:

```
SETDefault -SR Mode = PassiveBridging
```

By setting this parameter to PassiveBridging, all source-routed frames are transparently bridged across the spanning tree paths without examining or updating the routing information field (RIF) in the frame header. For information about the frame header, see "IEEE 802.5 Token Ring Frame Format Overview" later in this chapter.

- b Configure the same ring number on the bridge ports that are part of the same logical ring.

When you set up passive bridging, the same ring number must be assigned to all physical rings that are part of one logical ring.

For example, to create the logical ring 10 (decimal), on bridge 1 and bridge 2, enter:

```
SETDefault !1 -SR RingNumber = 10
SETDefault !2 -SR RingNumber = 10
```

- 2 Configure source route bridging for the remaining bridges outside the logical ring.

- a Configure ring numbers for the remaining bridges.

For example, to configure the ring numbers in decimal, on bridge 3 ports, enter:

```
SETDefault !1 -SR RingNumber = 10
SETDefault !2 -SR RingNumber = 30
```

- b Verify that IEEE bridging is enabled.

By default, the Mode parameter is set to IEEE. Verify its setting by entering:

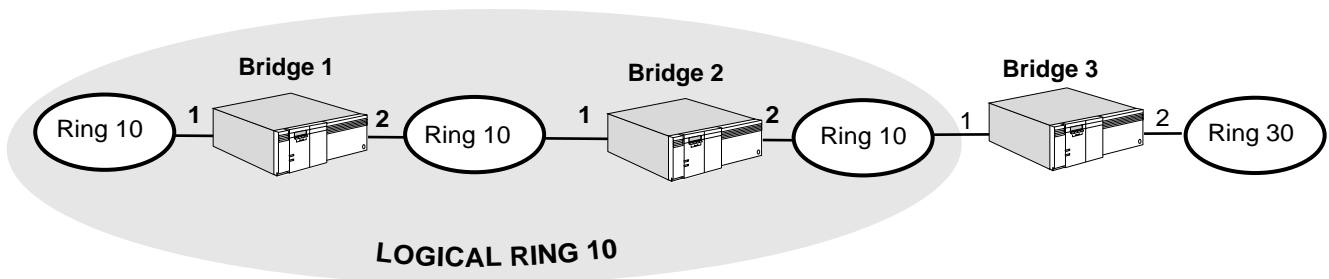
```
SHow -SR Mode
```

If the setting is not IEEE, configure this parameter by entering:

```
SETDefault -SR Mode = IEEE
```

By setting this parameter to IEEE, the forwarding path of the specifically routed frame (SRF) is determined by the RIF in the frame header. For information about the frame header, see "IEEE 802.5 Token Ring Frame Format Overview" later in this chapter.

Figure 42 Collapsing Multiple Rings into One Logical Ring with Passive Bridging



Setting Up Spanning Tree

In a source route bridging network, an end system can discover a route to a destination system on another ring by sending an All Routes Explorer (ARE) frame that is copied to every ring in the network. If only one path to the destination exists, the destination system only receives and responds to one ARE frame.

However, if multiple paths to the destination system exist, the destination system receives and responds to as many copies of the ARE frame as there are paths to it, resulting in heavy network traffic.

To limit the number of ARE frames in a source route bridging environment, 3Com bridge/routers can use the Spanning Tree Protocol (STP) to dynamically establish and maintain a spanning tree across all rings, allowing only a single spanning tree explorer (STE) frame to be forwarded on a ring and preventing duplicate ARE frames from appearing on the same ring. The STP Service is enabled by default so no additional user configuration is necessary. If the STP Service has been disabled, you can enable it by entering:

```
SETDefault -STP CONTROL = Enabled
```

You must disable transparent bridging on all ports before the STP packets are generated for the source route domain. Otherwise, the bridge/router generates STP packets for the transparent domain. To disable transparent bridging, use:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

Transparent bridging is not supported on model 32x and 52x SuperStack II bridge/routers. You do not need to perform this step for those bridge/routers.

For conceptual information about the Spanning Tree Protocol, see the Configuring Bridging chapter. For conceptual information about the route discovery process, see "Route Discovery Process" later in this chapter.

Configuring Parallel Bridges

This feature is not supported on model 32x and 52x SuperStack II bridge/routers.

If your network is composed of parallel source route bridges to provide redundancy as shown in Figure 36, you must assign unique bridge numbers to them using:

```
SETDefault -SR BridgeNumber = <number> (0-15) | 0x<number> (0-F)
```

3Com token ring bridges support both the decimal and hexadecimal format for the bridge number. Hexadecimal format entry must be preceded by a 0x.

As shown in Figure 36, bridge B has been assigned a bridge number of 4, and bridge E has been assigned a bridge number of 5.

Reducing Broadcast Traffic

You can reduce the amount of broadcast traffic in your source route bridging environment by regulating the maximum number of broadcast packets per second and setting the broadcast timer threshold to specify when to begin discarding broadcast packets.

To set the maximum amount of broadcast packets per second on a port, use:

```
SETDefault !<port> -BRidge BroadCastLimit = <packets per second>
(0-100000)
```

To set the broadcast limit timer threshold, use:

```
SETDefault -BRidge BLimitTimer = 400 | 600 | 800 | 1000 | Disabled
```

The broadcast limit mechanism works by counting the number of broadcast and multicast packets received during each timer interval. Broadcast and multicast

packets are forwarded during a timer interval until the broadcast limit threshold (described later in this chapter) for the port is reached. After the threshold has been reached, no additional broadcast or multicast packets are forwarded on the port until the start of the next timer interval. At that point, broadcast and multicast forwarding is resumed.

To disable the `BroadCastLimit` parameter, specify 0. To disable the `BLimitTimer` parameter, specify "Disabled."

Restricting Explorer Frame Propagation

You can restrict the propagation of ARE or STE frames to reduce unnecessary explorer traffic using:

```
SETDefault !<port> -SR MaxAreRDLimit = <number> (0-8)
SETDefault !<port> -SR MaxSteRDLimit = <number> (0-8)
```



Whether you use a value other than the default depends on your network configuration.

The `MaxAreRDLimit` parameter specifies the maximum number of route designators (RDs) (or hop count) allowed for an ARE frame received on the specified port. The default value of the `MaxAreRDLimit` parameter is eight, the maximum allowed in a source route bridging environment. This means that the maximum number of bridges or hops that can be daisy-chained in a source route bridge configuration is seven. You can further restrict the hop count by adjusting the `MaxAreRDLimit` parameter. When the source route bridge receives an ARE frame, it checks the setting of this parameter before forwarding it. If the setting is exceeded, the ARE frame is discarded.

The `MaxSteRDLimit` parameter specifies the maximum number of RDs allowed for an STE frame received on the specified port. The default value of the `MaxSteRDLimit` parameter is eight. If the number of route designators in the frame is equal to or greater than the `MaxSteRDLimit`, the frame is discarded. Otherwise, the STE frame is forwarded.

Configuring Filters

For complete information on configuring filters, see the *Configuring Mnemonic Filtering* chapter.

Configuring Security

You can use the bridge security features to select certain stations whose packets will be forwarded or blocked depending on their source or destination address. For complete information on using the `-BRIDGE SRcSecurity` and `DStSecurity` parameters, see "Bridge Security" in the *Configuring Bridging* chapter.

Configuring the Bridge/Router as an End System

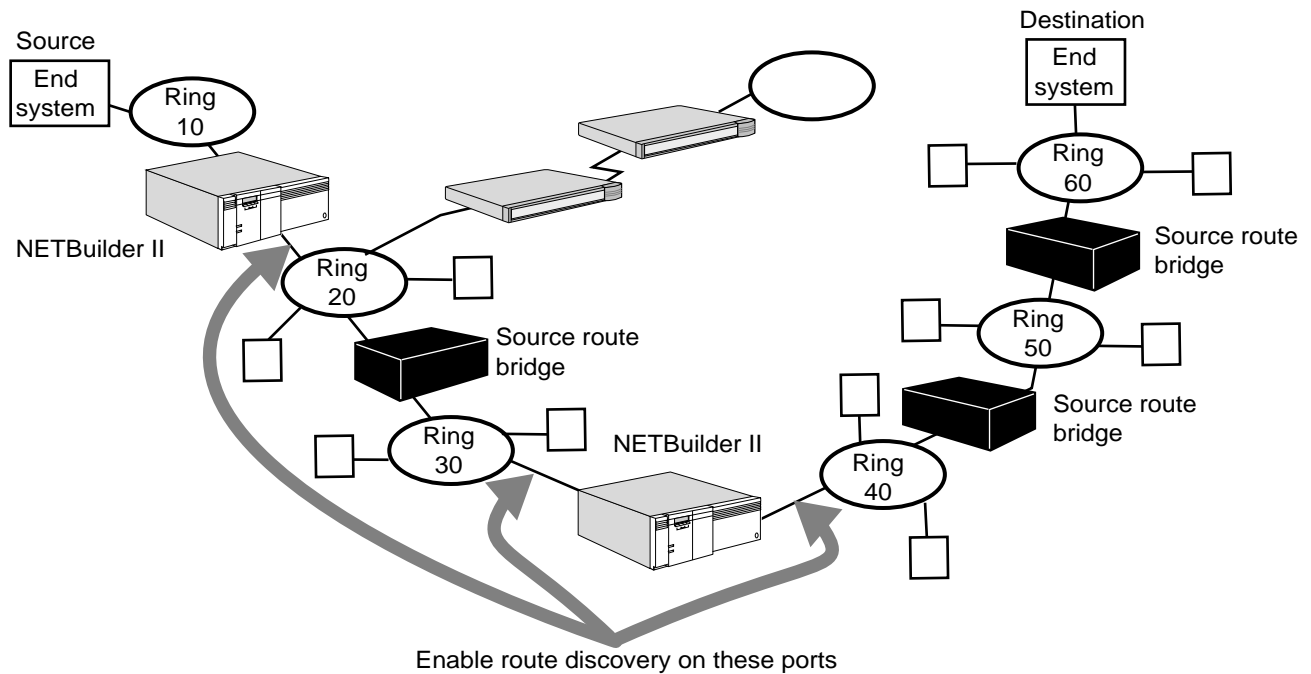
The remaining procedures in this section apply to the 3Com bridge/router functioning as an end system for network management purposes or as a level 3 router for routing protocol packets, such as Novell and AppleTalk in a source route environment.

Guidelines for Per-Port Route Discovery

You must configure route discovery on a port if the bridge/router must forward end system protocol packets to other end systems across a source route-only bridge. Use the following guidelines for setting `RouteDiscovery` in your network environment:

- Enable for IP on applicable ports if you want to network manage your bridges or routers that traverse a source route-only bridge. Enable route discovery for DLTest (Data Link Test) if you want to run DLTest to other 3Com bridges or routers that traverse source-route-only bridging environments.
- For routers in a source route-only environment as shown in Figure 43, enable RouteDiscovery for the appropriate protocols on applicable ports to ensure connectivity with source route-only end systems.

Figure 43 Route Discovery for Routers in a Source Route-Only Environment



For the following specific configurations, note the guidelines:

- Connecting a transparent domain with a source route domain
 Unless the SRTG feature is enabled, 3Com bridge/routers do not support the conversion of a transparent bridged frame to a source route bridged frame, or vice versa. You must configure a router to interconnect transparent domains (for example, connected through Ethernet ports) with source route domains (for example, connected through token ring ports). Enable Route discovery on ports that are connected to the source route-only domains or source route transparent domains with source route-only end systems.
- Routing domains connected by a source route-only bridged domain
 When you have two routing domains connected by one or more source route-only bridges, you must enable route discovery on the router ports directly connected to the source route-bridged domain.

Configuring Per-Port Route Discovery

To configure route discovery, use:

```
SETDefault !<port> -SR RouteDiscovery = ([All | None] | [AppleTalk | NoAppleTalk], [CLNP | NoCLNP], [DECnet | NoDECnet], [DLTest | NoDLTest], [IP | NoIP], [IPX | NoIPX] [LLC2 | NoLLC2], [VINES | NoVINES])
```

With this command, you can specify different combinations of protocols for end system route discovery to take place over a specific port.

The default for the RouteDiscovery parameter is None, which means that all end system packets are transmitted as transparent frames, and can reach end systems in a transparent bridged or a source route transparent (SRT) bridged environment.

You can specify that route discovery is initiated for all end system packets over a given port if a route to the destination end system does not exist in the local routing table. To specify route discovery for all end system packets, use:

```
SETDefault !<port> -SR RouteDiscovery = All
```



Specifying "All" can significantly impact the performance of the router. The router experiences a significant drop in the maximum packet forwarding rate during route discovery because of the additional CPU overhead required in route lookup and setup of the routing information of the packet. 3Com recommends that you enable RouteDiscovery only for the protocols you use. Increasing the value of the -SR HoldTime parameter will minimize the drop in forwarding rate for these protocols.

If you specify that route discovery is performed only for specific protocol types, you can enhance the performance for other protocols. For example, you can specify that over a given port, route discovery is performed only for AppleTalk and IPX packets using:

```
SETDefault !<port> -SR RouteDiscovery = (AppleTalk, IPX)
```

In this situation, all end-system packets that are not AppleTalk or IPX packets are transmitted as transparent frames over the port.

If the configuration changes, and you no longer want route discovery to take place for specific protocols, you can turn them off using the RouteDiscovery parameter. For example, to turn off route discovery for AppleTalk and IPX packets, use:

```
SETDefault !<port> -SR RouteDiscovery = (NoAppleTalk, NoIPX)
```

You can disable route discovery on a port using:

```
SETDefault !<port> -SR RouteDiscovery = None
```

For more information on end system source routing, see "How the Source Route Bridge Works" later in this chapter. For more information on the RouteDiscovery parameter, see *Reference for Enterprise OS Software*.

Discovering Routes to an End System

You can discover and optionally save a route to an end system using:

```
DiscoverRoutes <media address> [!<port>] [<timeout (1-120 sec)>]  
[AllRouteExp] [Xid] [Save]
```

where <media address> is [Cmac | Ncmac] %xxxxxxxxxxxx. x is a hexadecimal.

Use Cmac when <media address> is entered in canonical format and Ncmac for noncanonical input.



This command applies only to ports (token ring, FDDI, and HSS running Frame Relay, ATM, ATM DXI, SMDS, X.25, or PPP) with end system source routing enabled with the -SR RouteDiscovery parameter.

The media address should be preceded with the keyword Cmac or Ncmac for canonical or noncanonical format, respectively. The media address should also be preceded by a percent sign (%) and should be 12 hexadecimal digits.

All possible paths to the specified end system are displayed and a preferred route can be chosen and cached in the routing table.

For example, you can cause the bridge/router to issue a route discovery packet over port 1 to address %080000020003 in canonical format by entering:

```
DiscoverRoutes Cmac %080000020003 !1 30 Save
```

A response to the route discovery will be displayed in 30 seconds. If a route is found, the route traversed to reach the specified destination address is saved in the routing table. If one or more routes exist for the remote system, a prompt appears to request the preferred route to save and to determine whether the route is to be cached as a dynamic or a static route. After the route is saved, you can display it using the SHow -SR AllRoutes command.

For more information about the DiscoverRoutes command, see the Commands chapter in *Reference for Enterprise OS Software*.

Adding, Deleting, and Displaying Static Entries in the Routing Table

Routes to a destination end system are discovered using LLC TEST/XID frames. The route associated with the first TEST/XID response is cached in the routing table until its hold time expires. In some topologies, the route that is cached may not be the optimum route to the destination. Some end systems also cannot respond to TEST/XID frames. In these types of situations, you can configure the preferred route as a static (permanent) route using:

```
ADD !<port> -SR ROUTe <media address> [Override] [Dec | Hex] [<route>
[<largestframesize>]]
```

where:

<media address>	is [Cmac Ncmac] %xxxxxxxxxxx. 'x' is a hexadecimal. Use Cmac when <media address> is entered in canonical format and Ncmac for noncanonical input.
<route>	is <ring_number>&<bridge_number>[:<ring_number>] ...
<largest frame size>	is:
	0 for 516 bytes
	1 for 1,470 bytes
	2 for 2,052 bytes
	3 for 4,399 bytes
	4 for 8,130 bytes (not supported)
	5 for 11,407 bytes (not supported)
	6 for 17,749 bytes (not supported)

7 for 41,600 bytes (not supported)

For example, to configure a static route on port 2 of the bridge/router to the remote system with the MAC address %080002000001 and the manual override option (if the route configured for an end system address becomes invalidated for any reason, the static route is replaced by a learned route if one exists), enter:

```
ADD !2 -SR ROUTe Ncmac %080002000001 Override :55&1:56&2:57
```

To display the learned route associated with a specified end system in noncanonical and hexadecimal format, enter:

```
SHoW -SR ROUTe Ncmac %080002000001 Hex
```

To remove a static route from the routing table, you must remove it manually, unless you specified the Override option when you added the route. To remove a static route, use:

```
DELeTe !<port> -SR ROUTe <media address>
```

You can display routes from the routing table using:

```
SHoW [!<port> | !*] -SR ROUTe [[Cmac | Ncmac] %<media address>] [Dec | Hex]
SHoW [!<port> | !*] -SR AllRoutes [Dec | Hex] [<route>]
[Discover | Static] [<count>] <route>:
'<ring number>'&'<bridge number>.... | Transparent
SHoW [!<port> | !*] -SR WanRoutes
```

The SHoW -SR ROUTe command displays static routes in the routing table.

The SHoW -SR AllRoutes command displays dynamically discovered, static, and specific source routes or transparent routes depending on the options selected.

For example, to display all discovered routes in hexadecimal format off port 2 that have traversed bridge number 5, enter:

```
SHoW !2 -SR AllRoutes Hex &5 Discover
```

To display all static source routes in decimal format off port 2 that have traversed ring number 55, enter:

```
SHoW !2 -SR AllRoutes Dec :55 Static
```

To flush all discovered routes in hexadecimal format off port 1 that have traversed the partial route ring number 55, the bridge number 5, and the ring number 77, enter:

```
FLush !1 -SR AllRoutes Hex :55&5:77 Discover
```

The SHoW -SR WanRoutes command displays all learned remote networks (bridge number and ring number) and its associated data link connection identifier (DLCI), individual SMDS address, or X.25 DTE address for the Frame Relay, SMDS, or X.25 port, respectively.

For more information about these parameters, see the SR Service Parameters chapter in *Reference for Enterprise OS Software*.



If you have a static route in a source route environment, LLC will attempt to use that static route.

Aging Out Entries in the Routing Table

You can adjust the time interval (in minutes) that an inactive route entry can reside in the routing table using the HoldTime parameter. This parameter only affects the dynamically learned routes.

To change the default setting of 15 minutes, use:

```
SETDefault !<port> -SR HoldTime = <minutes>(1-1440)
```

Changing the Token Access Priority

The MinAccessPrior parameter determines the minimum access priority used for outgoing frames on a specified port. The lowest priority is 0; the highest is 6. End systems usually have a low-access priority, while bridges have a medium priority (the default is 4). You can configure a source route bridge that typically handles greater amounts of traffic to obtain the token more often than other end systems by adjusting the MinAccessPrior parameter.

To change the default setting of the MinAccessPrior parameter, use:

```
SETDefault !<port> -SR MinAccessPrior = <number>(0-6)
```

How the Source Route Bridge Works

This section provides conceptual information on the following topics:

- Source route, source route transparent bridging, and source route transparent bridging gateway (SRTG) definitions
- IEEE 802.5 token ring frame format
- Source route transparent bridging gateway concepts
- Route discovery process using ARE or STE frames
- End system source routing
- Routing tables

Definitions

This section provides definitions for source route bridging, source route transparent bridging, and source route transparent bridging gateway (SRTG).

Source Route Bridging

Source route bridging is supported on token ring, FDDI, and the following wide area networks: Frame Relay, ATM, ATM DXI, SMDS, X.25, PPP, and ISDN. Source route bridges connect token ring LANs and enable peer-to-peer and terminal-to-host communications across both LAN and WAN token ring networks.

When source route bridging is enabled, the bridge forwards packets based on a route determined by the end system from which the packet originated. The end system initiating the communication is responsible for dynamically determining and then maintaining information about the route to the destination. The source route information is contained within the frame and indicates the path through an extended network from the source to the destination. Because the end system and not the bridge determines the route, a bridge using source route bridging does not record or learn information about addresses on the surrounding networks in the same way that a transparent bridge does. The exception to this rule is on a Frame Relay, ATM DXI, SMDS, or X.25 interface, where the DLCI, VPI.VCI, SMDS, or X.25 address associated with the remote ring is learned.

Source Route Transparent Bridging

This feature is not supported on model 32x and 52x SuperStack II NETBuilder bridge/routers.

Source route transparent bridging is a combination of transparent and source route bridging. The bridge automatically determines whether a packet should be forwarded using transparent bridging or source route bridging. For example, if the bridge receives a frame with routing information, the bridge performs source route bridging. If the bridge receives a frame without routing information, it performs transparent bridging. Source route transparent bridging is used in topologies in which transparent end systems and source route-only end systems coexist on the same network; source route transparent bridging allows the transparent end systems to communicate with transparent end systems and source route-only end systems to communicate with source route-only end systems.

Source Route Transparent Bridging Gateway

This feature is not supported on model 32x and 52x SuperStack II NETBuilder bridge/routers.

With SRTG, you can connect a source-routed network to a transparent bridging network. The SRTG software provides a translation between source route and transparent bridging domains so that token ring network users can communicate with Ethernet network users using source routing; Ethernet network users can communicate using transparent bridging with token ring network users as though they were on the same LAN. Upon receipt of frames from a source route domain, SRTG translates them into transparent bridging frames and removes the source routing information fields (RIFs). The SRTG software also adds appropriate RIF fields to transparent bridging frames before forwarding them to a source route network.

You can configure your bridge to use transparent bridging only, source route bridging only, transparent and source route bridging simultaneously, or SRTG. When configuring parallel bridges, 3Com recommends that you configure both bridges in the same bridge mode, either source route or source route transparent, to prevent unexpected blocking of one type of traffic due to the Spanning Tree Protocol. For more detailed conceptual information about SRTG, see "Source Route Transparent Bridging Gateway Concepts" later in this chapter.

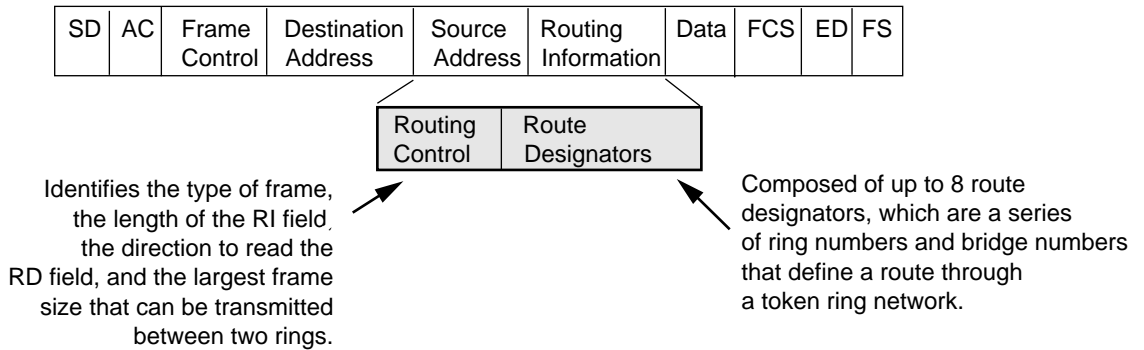
IEEE 802.5 Token Ring Frame Format Overview

Source route bridging requires that each end system in an extended network dynamically determines and maintains the routing information necessary to communicate with other end systems on remote rings in the network. Each frame transmitted by an end system contains the routing information a source route bridge needs to decide whether to forward the frame to an adjoining ring.

This section describes some of the fields in the IEEE 802.5 token ring frame (shown in Figure 44) that are important to a general understanding of the route discovery process. Only the destination and source address fields, as well as the

routing information field are discussed; not every field in an IEEE 802.5 token ring frame is discussed.

Figure 44 IEEE 802.5 Token Ring Frame Format



Destination address field

This 6-byte field identifies the end systems that are intended to receive and copy the frame.

Source address field

This 6-byte field identifies the system from which the frame originated. This field also contains a routing information indicator (RII) bit, which when set to 1, indicates the presence of the routing information field (RIF). If a source route bridge receives a frame with the RII bit = 1, it forwards the frame based on the routing information contained in the route designators (see the description of the routing information field). If a source route transparent bridge receives a frame with the RII bit = 0, it forwards the frame based on the destination address using the transparent bridging method.

Routing information field (RIF)

This 0- to 18-byte field contains routing control information and route designators (RD). The routing control information identifies, among other things, the type of source-routed frame, for example, an All Routes Explorer (ARE), Spanning Tree Explorer (STE), or specifically routed frame (SRF).

An ARE frame is transmitted by the source end system to every ring in the extended network. Because the ARE frame is forwarded by a source route bridge to every connected ring, the destination end system receives as many copies of the ARE as there are routes to it. ARE frames are originally transmitted with no route designators; as the frame is forwarded by source route bridges, route designators are added to the frame.

An STE frame is transmitted by the source end system and forwarded only by designated bridges, causing the frame to appear only once on every ring in an extended network. STE frames are originally transmitted with no route designators; as the frame is forwarded by source route bridges, route designators are added to the frame.

An SRF contains the specific route information that allows a source route bridge to forward the frame along a defined network path.

The RD field contains up to eight 2-byte route designators (route descriptors) of ring and bridge number information that describe the path to a destination.

Source Route Transparent Bridging Gateway Concepts

These concepts do not apply to model 32x and 52x SuperStack II NETBuilder bridge/routers.

The SRTG provides translation between source route and transparent bridging domains so that token ring network users can communicate using source routing with Ethernet network users, and Ethernet network users can communicate using transparent bridging with token ring network users. Upon receipt of frames from the source route domain, SRTG translates them into transparent bridging frames by removing the source route information fields (RIFs). SRTG adds appropriate RIF fields to transparent bridging frames before forwarding them to the source route network.

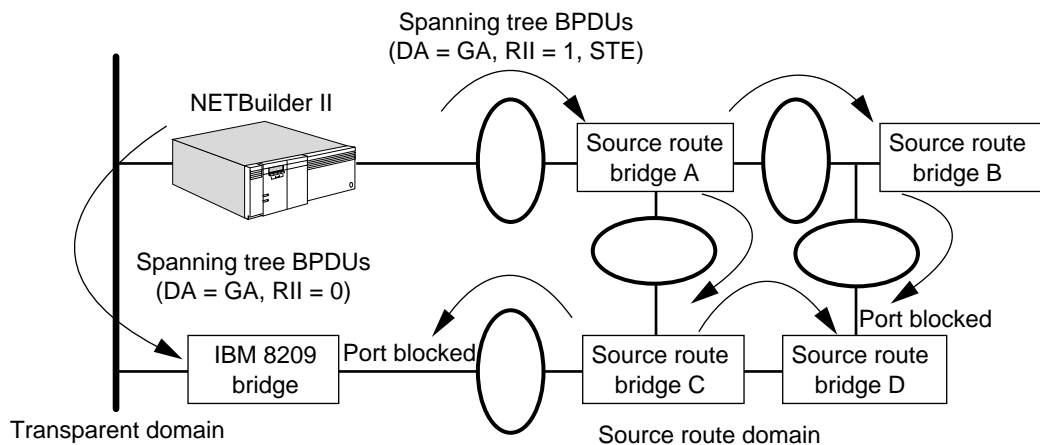
Spanning Tree Considerations

Two different spanning tree schemes exist for transparent bridging and for source routing. In transparent bridging, the Spanning Tree Protocol (STP) ensures that only one active path between any two stations exist in the network. In source routing, STP selects the bridges to forward the spanning tree explorer frames.

When both source route and transparent bridging domains are connected using SRTG, multiple gateways may be installed in parallel, either by mistake or on purpose, creating loops in the network topology. To eliminate loops and ensure a single active path between two stations, SRTG fully participates in the transparent STP.

To ensure compatibility with IBM 8209 or 8229 LAN bridges, the spanning tree entity on 3Com SRT gateways generates Bridge Protocol Data Units (BPDUs) as STE frames with a destination address set to the group address and the RII bit set. As shown in Figure 45, SRTG detects and breaks loops when there are multiple paths between SR and TB domains.

Figure 45 Spanning Tree Loop Detection by SRTG



Source route bridge A forwards the spanning tree BPDUs according to the source route spanning tree path to bridge B and C without recomputing the spanning tree algorithm. Bridge B forwards the BPDUs to bridge D, which drops them because the port is in the blocking state. Bridge C forwards BPDUs to bridge D and

the IBM 8209 bridge. Bridge D receives them but does not perform the spanning tree computation nor forward them because its other port is in the blocking state. When the IBM 8209 bridge receives the BPDUs, it detects a loop and blocks the source routing port.

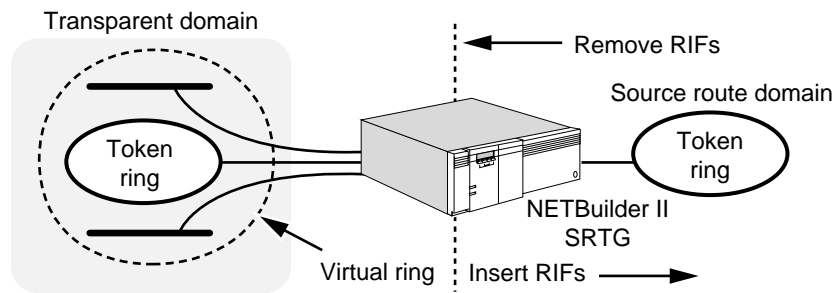
When the source routing port goes into a blocking state, all types of frames (ARE, STE, and SRF) are not forwarded. This behavior complies with the transparent bridging behavior but differs from pure source route bridging. When the primary and secondary SRT gateways change their role due to topology changes anywhere in the transparent bridging network (a primary gateway becomes secondary and vice versa), stations using the existing path may experience session disruption. Using parallel SRT gateways does not provide load balancing but does provide a backup path if the primary SRT gateway fails.

Packet Handling between Domains

When a packet is bridged from a source route domain to a transparent bridge domain using SRTG, the source route field of the frame is removed as shown in Figure 46. The RIF of the originator is cached with the direction bit in the route control field inverted for use by subsequent return traffic.

When a packet is bridged from a transparent bridge domain to a source route domain using SRTG, the packet is forwarded using the associated routing information from the source route table if the destination is known. If the destination is not known, the packet is immediately forwarded as an STE frame. The SRT gateway acts as a surrogate source routing station on behalf of all transparent bridge stations and uses a virtual ring number (set with the -SR GatewayVRing parameter) for its transparent bridge domain. Whenever bridging packets from the transparent bridge to source route domain, SRTG adds the virtual ring number and its own bridge number to the source route information of the destination station retrieved from the source route table. From the point of view of a source routing station, the entire transparent bridge LAN appears as a single source routed ring as shown in Figure 46.

Figure 46 Virtual Ring and Frame Translation



Source Route to Transparent Bridge Domain Packets. SRTG handles ARE, STE, and SRF frames as described in Table 11.

Table 11 Source Route to Transparent Bridge Domain Packet Handling

Frame Type	How Handled
ARE	An ARE frame with a group address (broadcast or functional) is forwarded onto the transparent bridge domain, but its source route information is not cached.

Table 11 Source Route to Transparent Bridge Domain Packet Handling (continued)

Frame Type	How Handled
	<p>An ARE frame with a specific destination address is forwarded onto the transparent bridge domain only when the destination address does not exist on the source route domain. If the source address is not found in the source route table, SRTG creates one. If the source address is found in the source route table, SRTG updates the old entry with the new route information if different.</p> <p>An ARE frame is copied as many times as available paths, and traverses all possible paths overriding the spanning tree configuration, causing SRTG to receive multiple copies of a packet if there are multiple paths. To prevent multiple copies from being forwarded to the transparent bridge domain, SRTG saves the source route from the first copy, considers it the optimal route, and discards the subsequent copies.</p>
STE	<p>An STE frame with a group (broadcast or functional) address is forwarded onto the transparent bridge domain, but route caching does not occur.</p> <p>An STE frame with a specific destination address is forwarded onto the transparent domain if the target station is not on the same source route domain. When forwarding a unicast STE frame, SRTG creates a new entry if an entry is not found in the source route table and the target station is known to exist on the transparent domain. If a source route entry already exists in the source route table, SRTG updates the entry but marks it as temporary. Because routes learned from STE frames may not be optimal, they are overwritten by any subsequent SRF frame from the source station.</p>
SRF	<p>Regardless of the destination address, SRTG forwards any SRF frame to the transparent bridge domain when RIF indicates the virtual ring.</p> <p>SRTG checks the ring out number and if it matches the SRTG virtual ring number, the SRF frame is translated and forwarded to the transparent bridge domain. If the destination station is already learned, the SRF frame is sent to a specific port. Otherwise, the SRF frame is flooded on all source route and SRTG ports except the source port. If no source route entry associated with the source station is found, SRTG creates one. If an entry is found but is temporary, SRTG updates the old entry and removes the temporary flag.</p>

Transparent Bridge to Source Route Domain Packets. When SRTG receives a packet from transparent bridge domain, it forwards the packet using the associated routing information from the source route table if the destination address exists in the database. If the destination does not exist in the source route table, SRTG immediately forwards the packet in a STE frame to reduce possible excessive traffic. Whether the destination address exists or not, SRTG adds the virtual ring number configured for the transparent bridging domain to the RIF field retrieved from the source route table.

Frame and Address Conversion

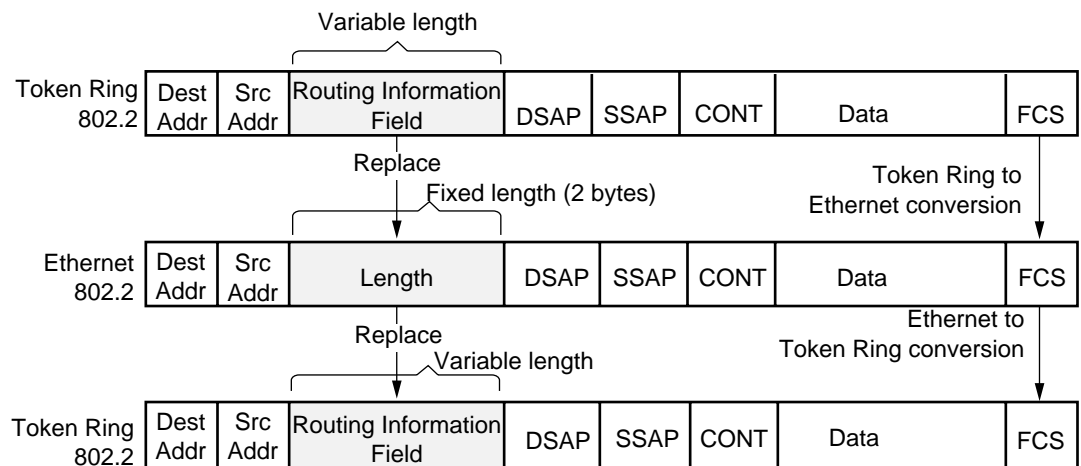
This section focuses on LAN-specific media (Ethernet, token ring, and FDDI) and the different packet formats. Frame conversions are necessary because Ethernet supports two different formats: Ethernet Version II frame and IEEE 802.3 frame.

In a source route token ring network, there are two ways to form a packet. IEEE 802.2 (LLC) encapsulation is used for LLC2 and NetBIOS packets while other protocols, such as IP, use SNAP encapsulation. To ensure compatibility with IBM's 8209 implementation of delivering bridged packets to a target station in its expected format, SRTG keeps track of the encapsulation format of each Ethernet station.

Ethernet 802.2 Conversion to and from Token Ring 802.2. Because both Ethernet and token ring supports IEEE 802.2 encapsulation, conversion of

Ethernet 802.2 frames to token ring 802.2 encapsulation is a simple task. SRTG removes the length field and adds the RIF field when it converts frames from Ethernet 802.2 to Token Ring 802.2. SRTG removes the RIF field and adds the length field (padding may be required for small frames) when it converts frames from Token Ring 802.2 to Ethernet 802.2. These frame conversions are shown in Figure 47.

Figure 47 Ethernet 802.2 Conversion to or from Token Ring 802.2

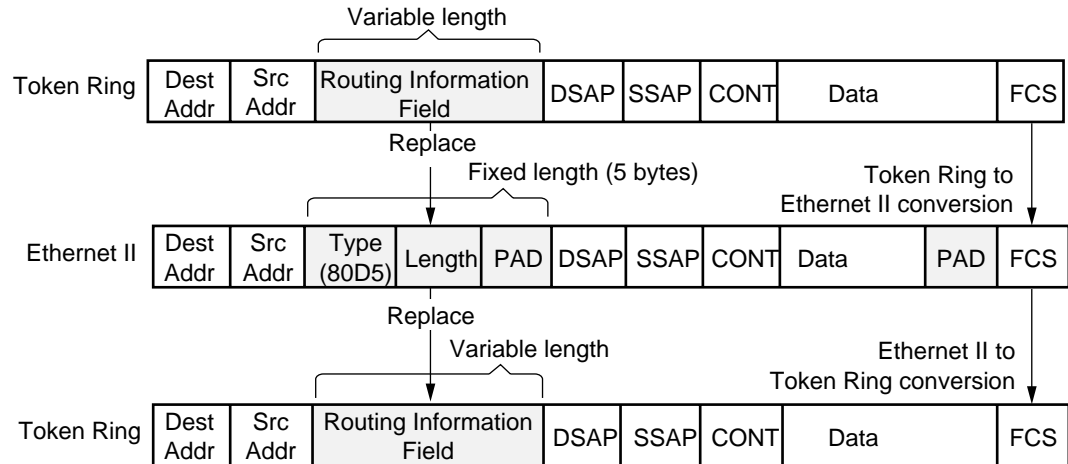


LLC-based Token Ring Conversion to and from Ethernet II To support the coexistence of both Ethernet II and Ethernet 802.3 frames on the same LAN, SRTG provides options in the -SR GatewayControl parameter to translate LLC-based packets to Ethernet II frames as follows:

- The destination service access point (DSAP) field in the token ring 802.2 frame must be a multiple of 4 (for example, 00, 04, 08, and so forth), except 0xBC and 0xE0, which are reserved for Banyan VINES and IPX, respectively.
- If SRTG is configured with the NoAutoMode setting, SRTG does not keep track of the encapsulation type of each transparent bridge station. The final encapsulation format is determined by the leeeMode or EtherMode setting of the GatewayControl parameter. If it is set to EtherMode, Ethernet II encapsulation with type 0x805D is used. If it is set to leeeMode, LLC-based packets are translated into the IEEE 802.3 format.
- If SRTG is configured with the AutoMode setting, different packet translation rules are used for known and unknown stations. For known stations, the leeeMode | EtherMode setting is ignored and the encapsulation format learned for each station is used. For unknown stations, LLC2-based packets are translated based on the leeeMode | EtherMode setting.

This resulting frame looks like an Ethernet II format. LLC data are not placed inside an 802.3 frame but placed into an Ethernet Version II frame whose type is specified as 0x80D5 and shown in Figure 48.

Figure 48 LLC-based Token Ring Conversion to and from Ethernet II



Maximum Frame Size

The maximum frame sizes used by Ethernet and token ring networks are different. To solve this frame length mismatch, SRTG automatically sets the largest frame size bit in the Route Control field to 1450 octets whenever it forwards frames to the token ring network (see Table 10). SRTG drops data packets from token ring or FDDI if the packets are larger than the Ethernet maximum frame size.

Route Discovery Process

An end system (PCs and workstations) with source route support installed can dynamically determine the routing information it needs to communicate with other end systems on remote rings interconnected by source route or source route transparent bridges. The route discovery process consists of the exchange of messages between the source and the destination end systems. Because no current standard for route discovery exists, the method that the end system uses may be protocol specific; therefore, a general description of the end system route discovery process is provided with details about how the 3Com bridge/router participates in the route discovery process.

The end station sends an explorer packet (for example, a TEST or XID frame, or a protocol-specific frame, in an ARE or STE) with the destination address in the header. If an ARE frame is transmitted by the source system and the source route bridge receives it, the source route bridge adds its bridge number and ring number of the adjoining ring to the RD fields, and forwards the frame to all of its source route bridging interfaces. The next source route bridge repeats the same process until the destination system recognizes its MAC address in the destination address field of the header and copies the frame.

If multiple paths to the destination system exist, the destination will receive as many explorer frames as there are paths and must respond to each explorer frame. The destination system responds to the ARE (each and every one) by sending an specifically routed frame (SRF). The frame contains all the routing information needed to forward the frame back to the source. In fact, when the source route

bridge receives an SRF, it forwards the frame according to the embedded source route information in the RD fields. When the source system recognizes its MAC address, it copies the frame and uses the routing information within the frame for all subsequent communications with that destination system.

If an STE frame is transmitted by the source system and the source route bridge receives it, the source route bridge adds its bridge number and ring number of the adjoining ring to the RD fields. The source route bridge only forwards the frame to the source route bridging interfaces that are not blocked because of the Spanning Tree Protocol, resulting in only one STE frame appearing on each ring. Each source route bridge in the spanning tree path follows the same procedure.

When the destination system recognizes its MAC address in the destination address field of the header, it copies the frame and responds to the STE by sending an ARE. The ARE frame is used so that all possible routes to the source can be found. On the return trip to the source system, the source route bridge forwards the ARE frame to all source route interfaces. When the source system recognizes its MAC address, it copies the frame (multiple responses may be received) and uses the routing information from the preferred ARE for all subsequent communications with that destination system.

When the 3Com bridge/router functions as an end system, it initiates the route discovery process by sending a TEST/XID STE frame. Upon receiving the frame, the destination system sends an ARE frame as described in the previous paragraph, except that the 3Com bridge/router caches the first ARE that it receives and discards all the other ARE responses.

End System Source Routing

Route discovery for end system source routing is supported on token ring and FDDI networks and wide area networks using PPP, Frame Relay, ATM DXI, SMDS, or X.25. Using end system source routing, the router acting as an end system can discover end systems not already present in the end system routing table. This is useful in situations in which the router receives a packet, but does not have a source route to the destination station on the source route network. If route discovery is enabled, the router determines the best route to the destination station by initiating a route discovery process and caching the discovered route in the routing table.

You normally use route discovery in configurations where the router is attached to a source route bridged environment. To enable routing in a source route environment, you must configure route discovery on the port directly connected to the source route bridged domain.

For any given port, you can configure the router to initiate route discovery for any combination of the following types of routing protocol packets:

- AppleTalk
- CLNP (OSI)
- DECnet
- IP (route discovery using an ARP packet)
- IPX
- VINES

You can also configure a port to initiate route discovery for DLTest packets.

Routes to a destination end system are discovered using LLC TEST/XID command frames. The route associated with the first TEST/XID response is cached in the routing table until its hold time has expired. Enabling route discovery allows end systems located in transparent-only, source route-only, or source route transparent environments to be reached.

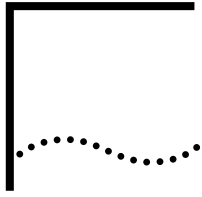
Routing Tables

You can access the routing table of end system by entering the SHow -SR AllRoutes command. For complete information on this parameter, see the SR Service Parameters chapter in *Reference for Enterprise OS Software*.

A source route bridge forwards a packet based on a route determined by the end system from which the packet originated. Routes are discovered on ports where the -SR RouteDiscovery parameter is enabled for one or more protocol packets. The routes learned by the bridge are cached in a routing table.

The two types of routing table entries are learned (dynamic) entries and user-assigned (static) entries.

- Learned (dynamic) entries are entries that the router learns from route discovery packets received from communicating end systems. The learned entries are subject to dynamic changes or deletion at intervals determined by the -SR HoldTime parameter (the default is 15 minutes).
- User-assigned (static) entries are entries assigned using the ADD -SR ROUTe command. The static entries can be changed or deleted only through the ADD or DELEte commands. These entries also are referred to as permanent entries.



CONFIGURING IP ROUTING

This chapter describes the procedures for configuring your system to perform Internet Protocol (IP) routing. It describes how the router works and gives guidelines for operating, managing, and troubleshooting it.



For conceptual information, see “How the IP Router Works” later in this chapter.

Configuring a Basic IP Router

The procedure in this section describes the minimum number of steps required to configure your system to route IP packets. Depending on your network requirements, you can use the default values of the parameters in the various services, or you can further configure the router according to later sections in this chapter.

To configure the IP router, you must set parameters in the RIP Service if your network uses the Routing Information Protocol (RIP) or in the OSPF Service if your network uses the Open Shortest Path First (OSPF) routing protocol. If you are using OSI routing for an IP environment, you must configure Integrated IS-IS parameters (IISIS).

The IP parameters enable the routing function and configure the networks connected to the router. The following information describes how to configure IP parameters.

Configuring for Local Area Networks and Point-to-Point Links

Use this procedure to configure basic IP routing over LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your router according to the information in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter and log on to the system with Network Manager privilege.
- Become familiar with the protocols supported by the router. This chapter describes the protocols only when the explanation is necessary for interpreting the parameters and screen displays used in the router software.
- Obtain an IP address for each port you want to configure (PPP links can be unnumbered).
- The router fully supports variable length IP subnetting. For information on IP addresses, subnets, subnet masks, and variable length subnet masks, see the Internet Addressing appendix. The router also supports multiple IP subnets. For more information on this feature, see “Configuring Multiple IP Networks/Subnets” later in this chapter.
- Obtain a subnet mask for the given network address, if it is different from the default or “natural” mask.

Procedure

To set up a basic configuration for your IP router, follow these steps:

- 1 Assign an IP address for each LAN port that will route IP using:

```
SETDefault !<port> -IP NETAddr = <IP address> [<subnet mask>
  [Ones | Zeros [MTU]]] | UnNumbered | IPCPAddress
```



CAUTION: An IP address assigned to port 0 is considered the IP address for all the interfaces. As a result, the bridge/router behaves as an IP host for Telnet access and network management and stops routing IP packets. Do not configure an IP address for port 0 if you want to route IP packets.

- 2 Assign an IP address or the value UnNumbered to each wide area port using PPP as the serial line protocol using:

```
SETDefault !<port> -IP NETAddr = <IP address> [<subnet mask>
  [Ones | Zeros [MTU]]] | UnNumbered | IPCPAddress
```

PPP does not require that you assign an IP address to each wide area port. Before configuring your IP router to route over PPP, determine if you want to assign an IP address to each wide area port. (See “Related Information” later in this chapter). If you do not want to assign an IP address to a wide area port, you must set the value of the -IP NETAddr parameter to UnNumbered. An advantage of not assigning an IP address to each wide area port is that you conserve valuable network and subnet numbers.

The NETAddr parameter lets you set the interface to be Internet Protocol Control Protocol enabled using the IPCPAddress parameter. The IPCPAddress parameter enables the bridge/router to automatically get an address from the internet service provider (ISP) and assign it to the local PPP WAN interface. After the IP address has been obtained, address mapping can take place.

- 3 If you are going to be running OSPF as the routing protocol over dial-up circuits, configure a demand interface circuit using:

```
SETDefault !<port> -OSPF DemandInterface = Enable
```



CAUTION: Do not configure any interface on any router in a single OSPF area as a demand circuit (DC) interface unless all routers in that area have been upgraded to at least software version 8.3.

With this setting, the router negotiates with the neighbor at the other end of the point-to-point link. If the neighbor agrees that the point-to-point link is a demand circuit, the router suppresses sending OSPF hello packets and routing refresh information, allowing the data link connection to be closed when not carrying application traffic. For the demand circuit to be cost-effective, make sure that it is isolated from as many topology changes as possible because topology changes bring up the interface.

For more information, see “Reducing Network Costs Using Demand Interface Circuits” later in this chapter.

- 4 Enable the dynamic routing protocols for IP routing using RIP, OSPF, or IISIS.
 - To enable RIP operation on a specified port, set the CONTROL parameter in the RIP Service (using its TALK and Listen values) as follows:

```
SETDefault !<port> -RIP CONTROL = ([Talk | NoTalk],
  [Listen | NoListen], [Poison | NoPoison], [TRigger | NoTRigger],
  [NetAdvUnn | SubnetAdvUnn], [SubnetBcast | AllIsBcast], [Aggregate |
  NoAggregate], [DeAggregate | NoDeAggregate], [DynamicNbr |
  NoDynamicNbr],
  [FullMesh | NonMesh])
```

Setting the CONTROL parameter to the TALK and Listen values enables the router to send and receive routing information with other routers using RIP.

You can also configure RIP for networks with variable length subnet masks using an aggregate/deaggregate scheme or the range table mask scheme. For more information, see “Configuring RIP for Networks with Variable Length Subnet Masks” later in this chapter.

If you set the value of the -IP NETaddr parameter to UnNumbered for a PPP serial link, make sure that you set the value of the -RIP CONTROL parameter to NetAdvUnn or SubnetAdvUnn depending on your network configuration.

- To enable OSPF on a specified port, set the CONTROL parameter in the OSPF Service using:

```
SETDefault !<port> -OSPF CONTROL = Enable
```

After OSPF is enabled, the router will exchange routing information with other routers using OSPF.

- To configure IISIS for Dual IP and Open System Interconnection (OSI) mode, enter:

```
SETDefault -IISIS CONTROL = Enable
```

- 5 Enable IP routing by entering:

```
SETDefault -IP CONTROL = RRoute
```

To complete the configuration for PPP links, see the Configuring Wide Area Networking Using PPP chapter.

Related Information

A serial line running PPP can support IP routing without the assignment of IP subnets. This feature is called *unnumbered links*. An unnumbered PPP link is useful only between two routers; in other words, it cannot connect a router to a host.

You must configure a serial line running PPP as an unnumbered link using the -IP NETaddr parameter before the unnumbered link takes effect. When an update is sent over an unnumbered PPP link, the source IP address is borrowed from another interface. For this reason, a router must have at least one IP address configured.

When RIP is run over a PPP link, both ends of the link must be either unnumbered or numbered with the same IP subnet. Half-numbered links, or links with inconsistent IP subnets on both ends, are considered a configuration error.

When OSPF or IISIS is run over unnumbered PPP links, no limitation exists in the way that the PPP link may be configured. Either end of the link can be numbered independently, or both ends can remain unnumbered. If both ends are numbered, they need not be on the same IP subnet nor have the same subnet masks.

You do not need to assign a network number to a Frame Relay cloud if you are using IISIS.

Configuring for Wide Area Networks

IP routing over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), X.25, and ATM is supported over fully meshed, partially meshed, and nonmeshed topologies.

If you plan to use RIP over Frame Relay, ATM DXI, X.25, or ATM in a partially meshed or nonmeshed topology, you must enable the next-hop split horizon feature by having a list of neighbors and you must set `-RIP CONTROL` to `NonMesh`. The list of neighbors can be dynamically generated by the system or manually configured.

If you plan to use OSPF over Frame Relay, ATM DXI, X.25, or ATM in a partially meshed or nonmeshed topology, you can create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. You must run the OSPF `NonMesh` mode over the Frame Relay, ATM DXI, or X.25 cloud.

If you plan to use ISIS over Frame Relay, ATM DXI, or X.25 in a partially meshed or nonmeshed topology, no additional configuration is necessary. Regardless of the type of topology, when you use ISIS, you do not need to assign a network number to the Frame Relay, ATM DXI, or X.25 cloud.

Routing IP over Switched Multimegabit Data Service (SMDS) is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your IP router to perform routing over SMDS, see the *Configuring Wide Area Networking Using SMDS* chapter.

For information on configuring PPP, see the *Configuring Wide Area Networking Using PPP* chapter. For information on wide area networking using ISDN, see the *Configuring Wide Area Networking Using ISDN* chapter.

Verifying the Configuration

To verify the configuration, examine network devices and send packets from one network to another using the PING command.

Examining Network Devices

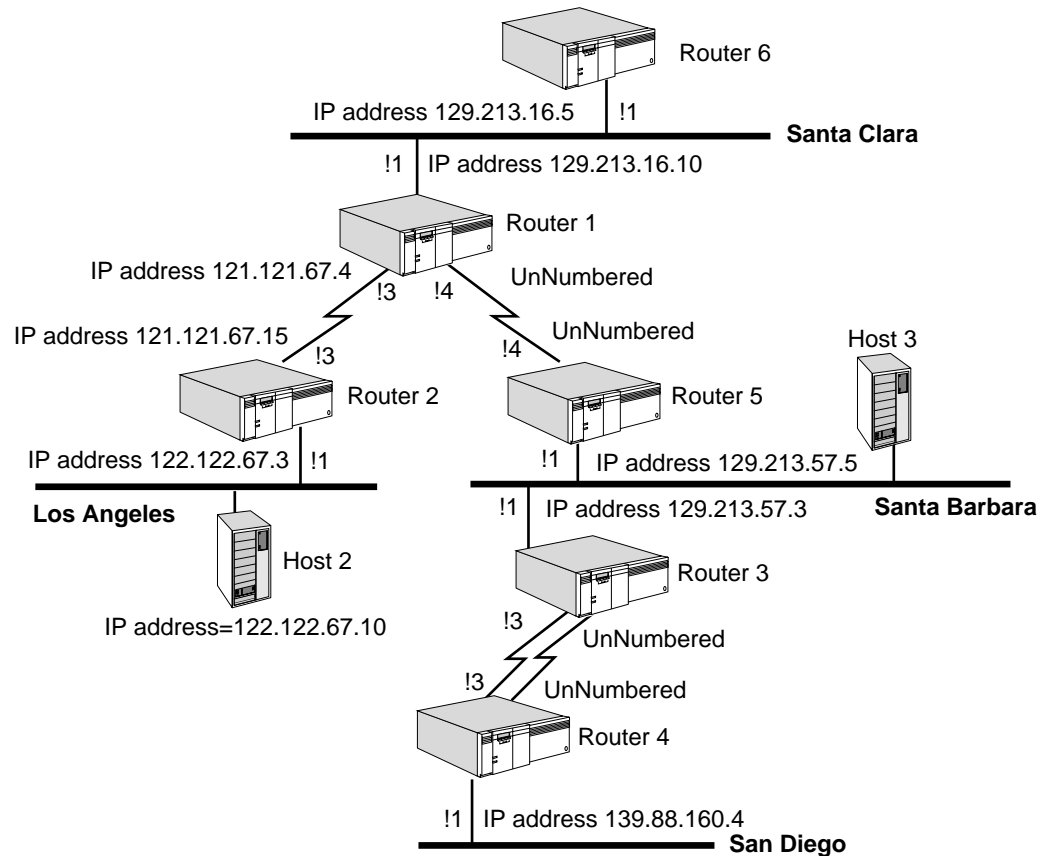
To examine the status of the IP router, follow these steps:

- 1 Display information on the attached networks by entering:
`SHOW -IP NETaddr`
- 2 Determine which stations or networks are reachable from the router by entering:
`SHOW -IP AllRoutes`
- 3 Display information from the Address Translation Table by entering:
`SHOW -IP ADDRESS`

Checking with PING

To use the PING command to check if packets are being forwarded, use Figure 49 as an example, and follow these steps:

Figure 49 Wide Area Router Configuration



- 1 Determine whether all of the router 1 network interfaces are up and running by entering the following commands from router 6:
 - a To check port 1, enter:


```
PING 129.213.16.10
```

If you do not specify the amount of time in seconds that the bridge/router should attempt to ping a device, the bridge/router assumes 20 seconds. For more information on the PING command, see the Commands chapter in *Reference for Enterprise OS Software*.
 - b To check port 3 of router 1, enter:


```
PING 121.121.67.4
```

If a port is operational, a message similar to the following appears after each PING command:

```
pinging ... 121.121.67.4 is alive
```

If a port is not operational, a message similar to the following appears:

```
pinging ... 121.121.67.4 is not responding
```

If this message appears, check the network connection to see if the cables are properly connected. Contact your network supplier or 3Com for help if you still cannot determine the cause of the problem.
- 2 After you determine that each port is operational, check that the router can forward packets from one network to another.

- a Enter the PING command on router 6 to check if it can communicate with host 2:

```
PING 122.122.67.10
```

If host 2 is operational and the router functions properly, a message similar to the following appears:

```
pinging ... 122.122.67.10 is alive
```

- b If you do not get this message, use the TraceRoute command on router 6 to trace a path to your intended destination. Specify the IP address of the destination you want to trace.
- c Follow the steps in “Checking the Overall Status” to verify the following items:
- Port 1 is properly configured on router 1.
 - Port 3 is properly configured on router 1.
 - Port 1 is properly configured on router 2.
 - Port 3 is properly configured on router 2.
 - The address used in the PING command is the correct address of host 2.
 - Routing is enabled on router 1, router 2, and router 6.
 - The routing protocol is properly configured.

If you cannot determine the cause of the problem, contact 3Com or your network supplier for help.

Getting Statistics

After you have followed the necessary setup and checking procedures using the PING command, examine the statistics by entering:

```
SHow -SYS STATistics -IP
```

You can collect statistics for a specific period by using the SampleTime and STATistics parameters. For more information, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*.

Checking the Overall Status

The following information pertains to checking the status of the router.

Procedure

To check the overall status of the IP router, follow these steps:

- 1 Examine the path configurations by entering:

```
SHow -PATH CONFIguration
```
- 2 Examine the port configurations by entering:

```
SHow -PORT CONFIguration
```
- 3 Examine the IP configurations by entering:

```
SHow -IP CONFIguration
```
- 4 Examine the RIP configurations by entering:

```
SHow -RIP CONFIguration
```
- 5 Examine the OSPF configurations by entering:

```
SHow -OSPF CONFIguration
```
- 6 Examine the ARP configurations by entering:

SHoW -ARP CONFIguration

- 7 Examine the BGP configurations by entering:

SHoW -BGP CONFIguration

- 8 Examine the ISIS configurations by entering:

SHoW -ISIS CONFIguration

Related Information

You may also want to verify that routing protocols and static routes are configured properly by using the TraceRoute command. For example, at router 1 in Figure 49, you can trace the route between routers 1 and 4, which will verify that routers 5 and 3 relayed the packets sent by router 1. For complete information on the TraceRoute command, see the Commands chapter in *Reference for Enterprise OS Software*.

Customizing the IP Router

After you set up and check the configuration of the basic IP router, it is ready to perform packet routing. If desired, you can further customize your IP router by doing the following tasks:

- Configure UDP Broadcast Helper, if necessary
- Configure multiple subnets
- Configure logical networks over IP
- Configure RIPv2 for networks with variable length subnet masks
- Configure static routes
- Configure packet filtering
- Configure routing policies
- Use the IP security parameters
- Configure interautonomous system routing using the Border Gateway Protocol (BGP)

Configuring UDP Broadcast Helper

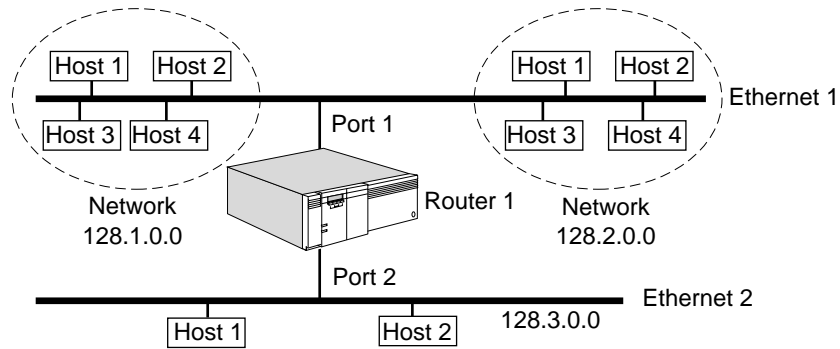
UDP Broadcast Helper allows applications in the TCP/IP stack to forward broadcast packets through a gateway and to another network segment. For information on configuring UDP Broadcast Helper, see the Configuring UDP Broadcast Helper chapter.

Configuring Multiple IP Networks/Subnets

Your IP router supports multiple IP subnets. You can configure more than one IP network or subnet on any media.

The procedure for configuring multiple IP subnets on all interfaces is the same. The following paragraphs present an example of configuring multiple IP subnets on an Ethernet network.

Figure 50 is an example of a topology where two IP networks can be configured on an Ethernet. The first IP network is called 128.1.0.0; the second is called 128.2.0.0. Use the following example to configure these IP networks on Ethernet 1. Configure the two networks on port 1 of router 1.

Figure 50 Two IP Subnets Configured on the Same Ethernet

To configure the two networks on port 1 of router 1, follow these steps:

- 1 Configure the first network.

The first address that you configure is known as the primary address. This address is indicated by an asterisk when you enter the `SHoW -IP NETAddr` command. For example, to set up network 128.1.0.0 with the IP address of 128.1.0.5, enter:

```
SETDefault !1 -IP NETAddr = 128.1.0.5
```

- 2 Configure any subsequent networks.

To configure a subsequent network, for example, network 128.2.0.0 with the IP address of 128.2.0.5, enter:

```
ADD !1 -IP NETAddr 128.2.0.5
```

To delete an address, use:

```
DElete !<port> -IP NETAddr <IP address>
```

In the topology shown in Figure 50, the systems on Ethernet 1 have been divided into two IP networks. Direct communication takes place among the hosts in network 128.1.0.0 and among the hosts in network 128.2.0.0. However, router 1 must forward packets between a host on network 128.1.0.0 and a host on network 128.2.0.0.

Related Information

The ability to configure multiple IP subnets gives you the following advantages:

- You can maximize the use of your network media.

The structure of the IP address limits the number of systems that can be addressed on an IP network. Configuring multiple IP subnets on a single network media allows you to increase the number of systems that you can address on a single media.
- You can break down systems on the network media into subsets of virtual private networks (VPNs). Direct communication occurs within these VPNs.

You must factor the advantages of being able to configure multiple IP subnets against the fact that traffic on a segment containing multiple IP subnets can increase significantly. Traffic from one network to another on the same segment must first go to the router then back out on the same segment.

Configuring Logical Networks over IP

You can assign the same IP address to several ports, and bridge among those ports while routing to other ports, by creating multiple logical networks (MLN).

MLN offers the following benefits for IP routing:

- Simplifies network protocol address administration on large networks. Instead of configuring each port individually, you need to configure only the group port.
- Reduces the number of IP addresses you need, making more efficient use of the limited hierarchical IP address space.
- Allows you to move stations from one LAN to another LAN without having to reassign hierarchical IP addresses, as long as both LANs belong to the same logical network.
- Allows you to integrate a number of bridged networks by routing them from the bridged environments (configured as logical networks) across a LAN or WAN backbone.
- Allows you to restrict broadcasts by grouping the target range into a port group since bridging of a logical network occurs only within the port group.

In Figure 51, ports 1, 2, and 3 and the LANs attached to them have been grouped together into logical network V1 by entering:

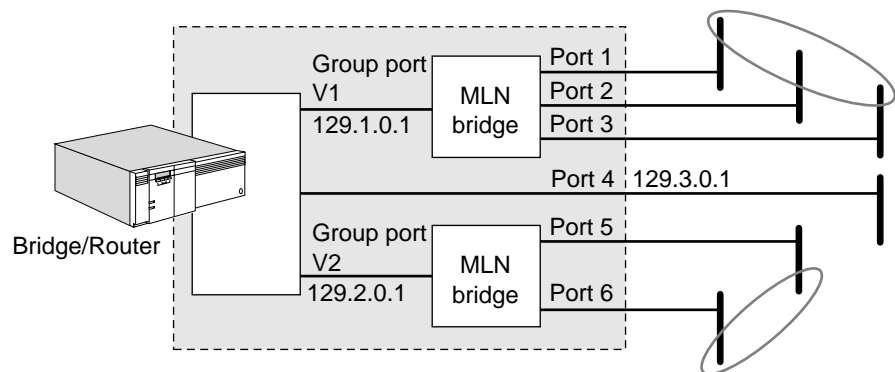
```
ADD !V1 -PORT LogicalNET EThernet 1,2,3
```

Ports 5 and 6 have been grouped into logical network V2 by entering:

```
ADD !V2 -PORT LogicalNET EThernet 5,6
```

Port 4 is an ordinary port that does not belong to a logical network.

Figure 51 Logical Networks over IP



For more information about configuring group ports, see "Configuring Multiple Logical Networks" in the Configuring Advanced Ports and Paths chapter.

You can now assign IP addresses to ports V1, V2, and 4, for example:

```
SETDefault !V1 -IP NETaddr = 129.1.0.1
SETDefault !V2 -IP NETaddr = 129.2.0.1
SETDefault !4 -IP NETaddr = 129.3.0.1
```

You cannot assign IP addresses directly to the member ports 1, 2, 3, 5, and 6.

You can also assign subnet masks and enable dynamic routing protocols, as explained in "Configuring a Basic IP Router" earlier in this chapter, or customize the router in the other ways explained in this chapter. Ports 1, 2, and 3 share the IP

address and other IP properties that you assign to port group V1, and ports 5 and 6 share the IP address and other IP properties that you assign to port group V2.

With this configuration, traffic between any port in group V1 and any port in group V2 is routed. Traffic between group V1 and port 4, or between group V2 and port 4, is also routed.

Traffic among ports within a port group is bridged, not routed. Traffic for the network protocol configured on group port V1 is bridged among ports 1, 2, and 3 (as indicated by the MLN bridge in the figure). Traffic on group port V2 is bridged between ports 5 and 6. To configure this bridging, you must enable global bridging and per-port transparent bridging on all member ports. For more information, see “Bridging over Multiple Logical Networks” in the Configuring Bridging chapter.

Adding a Static IP Address

When you add a static IP address for a group port using the ADD -IP ADDRESS parameter, and the group port has a member in one of the 6-port Ethernet cards, then the IP traffic will not be forwarded to those member ports. However, if the MAC address is learned dynamically through the Address Resolution Protocol (ARP) on those member ports, then traffic is forwarded.

Configuring RIPv1 for Networks with Variable Length Subnet Masks

You can use variable length subnet masks in your network and use RIPv1 as the routing protocol. By using variable length subnet masks, you can eliminate the need for additional IP addresses, which are increasingly difficult to obtain because of the rapid growth of IP-based networks.

Because RIPv1 packets do not carry explicit subnet information, you can configure the router that receives or transmits a routing update to determine the appropriate subnet mask. You can implement the following schemes:

- **Aggregate/deaggregate scheme**
The aggregate/deaggregate scheme primarily addresses the transmitter function and how the transmitting router translates routes from one mask length to another so as not to confuse receiving routers.
- **Range table mask scheme**
The range table mask scheme addresses the receiver operation and how the receiving router interprets an incoming route advertisement and assigns an appropriate subnet mask to it.

These two schemes are independent of each other; they can be used individually or together.

The routes can be RIP-learned routes, directly attached networks, static or dynamic routes learned from other protocols (if the appropriate policy is enabled).

Using the Aggregate/Deaggregate Scheme

To enable the aggregate/deaggregate scheme on a specified (outgoing) port, use:

```
SETDefault !<port> -RIPIP CONTROL = Aggregate
SETDefault !<port> -RIPIP CONTROL = DeAggregate
```



Do not use the aggregate/deaggregate scheme with unnumbered PPP links. Use the SubnetAdvUnn / NetAdvUnn values with the -RIPIP CONTROL parameter or use the range table mask scheme.

Procedure When both values are selected, RIPv2 performs route conversion using the following algorithm.

To use the Aggregate/Deaggregate scheme, follow these steps:

- 1 If it is a host route (with mask 255.255.255.255), propagate as is.
- 2 If it is a network route using the natural mask (for example, 10.0.0.0 255.0.0.0), propagate as is.
- 3 If it is a subnet route, do the following:



When NoAggregate is selected, step 3b is skipped. When NoDeAggregate is selected, step 3c is skipped.

- a If the outgoing interface is not subnetted, or if the outgoing interface belongs to a different IP network number, then zero out the bits in the subnet portion and propagate the natural IP network number (aggregate to the natural mask).

For example, if the routing update is sending 10.1.0.0 to 11.1.0.0, then the subnet portion is zeroed out and the natural IP network number (10.0.0.0) is propagated.

- b If the outgoing interface has the same IP natural network number as the route being propagated and if the mask of the route is longer than the mask of the outgoing interface, adopt the shorter mask and zero out all the bits in the host field (aggregate to a shorter mask).

This step of the algorithm is used if Aggregate is selected on the outgoing port. For example, if the mask of the route is 255.255.255.0 and the mask of the outgoing interface is 255.255.0.0, the shorter mask is used and the host field bits are zeroed.

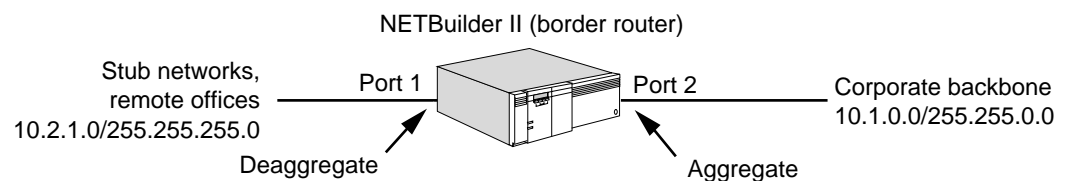
- c If the outgoing interface has the same IP natural network number as the route being propagated and if the mask of the route is shorter than the mask of the outgoing interface, adopt the longer mask and convert the route into a series of route advertisements that cover the full address space.

This step of the algorithm is used if DeAggregate is selected on the outgoing port. For example, if the route 10.1.0.0 255.255.0.0 is sent to neighbors with a longer mask (255.255.255.0), the route is expanded into a sequence of subnets 10.1.0.0 255.255.255.0 to 10.1.255.0 255.255.255.0. In this example, each route from the shorter side translates into 256 routes on the longer side.

- d If a, b, and c are not true, propagate as is (the outgoing interface is subnetted from the same IP network as the receiving interface and has the same mask).

Related Information Use the aggregate/deaggregate scheme in simple network topologies; for example, you may have a single router between the corporate backbone and stub networks (or remote offices) as shown in Figure 52.

Figure 52 Route Aggregation/Deaggregation with RIPv2



In this configuration, the backbone has a shorter mask and no overlapping routes (10.2.0.0 255.255.0.0 and 10.2.2.0 255.255.255.0 are overlapping and cannot coexist). All subnets with the same aggregate must be fully connected and contiguous; subnets with different aggregates can be located independently. For example, all 10.2.X.X subnets must be connected and contiguous, but 10.2.X.X and 10.3.X.X can be independent.

The aggregate/deaggregate scheme provides the following benefits:

- The aggregate scheme reduces the number of routes in the backbone routing table; a smaller routing table leads to smaller routing overhead and smoother network operation.
- All the work is performed by the border router that has interfaces to subnets with different masks. It is responsible for translating routing updates and using the correct subnet mask. No other routers need to be aware that variable length subnet masks are used.
- The scheme can be used easily and quickly in a simple network topology as shown in Figure 52 with minimal impact; it is compatible with older routers and major upgrades are not necessary.

Using the aggregate/deaggregate scheme has the following disadvantages:

- Deaggregation may not be a beneficial scheme in some topologies (do not use it for unnumbered PPP links or topologies more complex than shown in Figure 52); the default route advertisement should be used as an alternative.
- If more than one router connects the longer subnets to the backbone, the RIP advertisements, after aggregation or deaggregation, may confuse each other. You may need to configure the `-RIPIP ReceivePolicy` parameter to filter out this type of information.

Using the Range Table Mask Scheme

To configure the range table mask scheme, use:

```
ADD -RIPIP RcvSubnetMask <IP address> <IP address> <subnet mask>
```

For example, you could specify that all subnets between 10.2.0.0 and 10.2.255.0 use subnet mask 255.255.255.0 by entering:

```
ADD -RIPIP RcvSubnetMask 10.2.0.0 10.2.255.0 255.255.255.0
```

You can configure any type of subnet mask to any network number. The subnet mask can be longer or shorter than its natural mask. For example, the range 128.4.0.0 128.4.0.0 with subnet mask of 255.252.0.0 means that network 128.4.0.0 is assigned 255.252.0.0 as the subnet mask.



You cannot assign a subnet mask to the default route (0.0.0.0).

Procedure For each route received from a neighbor, RIPIP determines the appropriate mask using the following algorithm:

- 1 If the route belongs to the same IP network number of the receiving interface, use the mask of the interface.

For example, if the router receives route 10.1.0.0 on interface 10.2.0.0 255.255.0.0, the router adopts the same subnet mask for the received route.

- 2 If the route belongs to the same IP network number of any other interface, adopt the mask of that interface.

This action is useful when receiving routes over an unnumbered PPP link.

- 3 Use the natural mask based on the class (A, B, or C) of the received route.

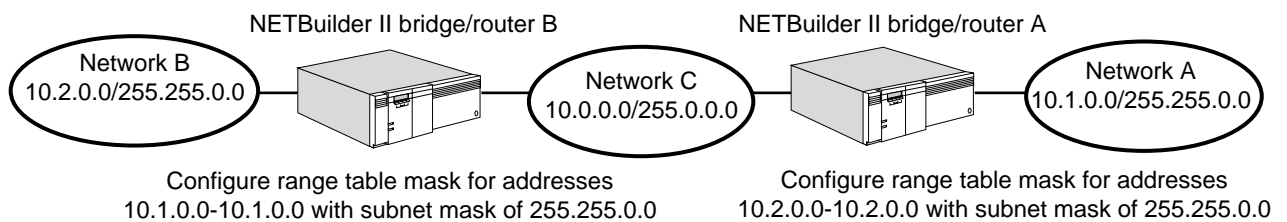
After the mask has been determined, the software checks to see if the route falls within any of the network ranges in the range table. If there is a match, the software overrides the mask with a user-configured subnet mask. If there are multiple matches in the table, the software picks the most specific match. The entries in the range table are organized from the more specific to the less specific. For example, range 10.1.0.0–10.1.255.0 with a mask of 255.255.255.0 is more specific than 10.0.0.0–10.255.0.0 with a mask of 255.255.0.0. With these ranges in the table, network 10 is assigned subnet mask 255.255.0.0; networks that begin with 10.1.X.X are assigned a subnet mask of 255.255.255.0.

As the final step, the software compares the received route with the mask. If there are non-zero bits in the host field, the route is a host route, and the software converts the subnet mask to 255.255.255.255.

Related Information Use the range table mask scheme for more complex topologies not covered by the aggregate/deaggregate scheme (overlapping routes exist, routes learned over unnumbered PPP links). With the range table mask scheme, there is no limit on the number of potential subnet masks.

The topology in Figure 53 has overlapping routes. In this situation, you need to configure the range table mask on NETBuilder bridge/routers A and B. You prevent bridge/router A, for example, from adopting the shorter mask (255.0.0.0) of network C when receiving route updates from network B.

Figure 53 Range Table Mask



The range table mask scheme provides the following benefits:

- Classless addressing can be supported.
- No topology limitations exist.
- Overlapping routes are supported.

Using the range table mask scheme has the following disadvantages:

- Extensive configuration may be required on every router, leading to increased administrative overhead.
- When the scheme is first used, all routers must be upgraded and synchronized at the same time. Because all routers must be configured with identical information, the coexistence of non-NETBuilder bridge/routers may not be possible.

Adding RIP V2 Compatibility

RIPV2 routers can inter-operate with RIPv1 routers. To facilitate this, as well as to offer a migration path for existing routers, RIPV2 capable routers have several modes of operation. Four mode settings are available for transmit:

- RIPv1 Only – Only RIPv1 messages are sent.
- RIPv1 Compatible – RIPv2 messages are sent using directed broadcast if no AdvToNeighbor is set. Otherwise, the RIPv2 messages are sent to the neighbors individually using unicast. Note that these RIPv2 messages are picked up by RIPv1 routers.
- RIPv2 Only – RIPv2 messages are sent using multicast address 224.0.0.9 if no AdvToNeighbor is set. Since RIPv1 routers do not listen on that multicast address, they do not pick up the RIPv2 messages.
- None – No RIP message is sent.

Four mode settings are available for receive:

- RIPv1 Only – Only RIPv1 messages are received.
- RIPv1 Compatible – Both RIPv1 and RIPv2 messages are received.
- RIPv2 Only – Only RIPv2 messages are received.
- None – No RIP message is received.

In the brouter implementation, sets of modes for sending and receiving are available and are provided on a per interface basis. The None mode (where sending/receiving of RIP messages are stopped) is obtained by setting the corresponding IPRIP Control parameter to NoTalk/NoListen. You can set both the send and receive modes to the same value.

RIPv1 mode is the default mode of operation, which avoids problems with RIPv1 routers that may have trouble interpreting a RIPv2 message (although a properly implemented RIPv1 router would not have this problem). When the mode is set to RIPv1 Compatible, the router can exchange RIP information with both RIPv1 and RIPv2 routers. However, in the RIPv1 Compatible mode it is sending RIPv2 message to RIPv1 routers, so some issues involving subnet arise.

For more information about the V2CompatMode parameter, see the RIPIP Service Parameters chapter in *Reference for Enterprise OS Software*.

Adding Authentication

Every RIPv2 message can contain an optional authentication block. The authentication block occupies the same size as a route entry (20 bytes) and is distinguished by a value of 0xFFFF in the Address Family Identifier field. If the authentication block is present, it must be immediately following the RIP header, and there must only be one such authentication block in the entire RIPv2 message. Currently, only the authentication type of 0x2 is specified, which denotes that the next 16 bytes contain the authentication string (password), expressed in ASCII characters. If the authentication string is less than 16 bytes, it is padded to the right with null. For more information about the V2AUTHenticate parameter, see the RIPIP Service Parameters chapter in *Reference for Enterprise OS Software*.

Adding Authentication Password

RIPv2 introduces the ability to authenticate a RIPv2 message. However, the recipient has a choice of ignoring this password authentication if it so chooses. Consequently, the rules for accepting a RIPv2 message varies depending on the authentication setting of a router.

For more information about the V2PassWord parameter, see the RIPIP Service Parameters chapter in *Reference for Enterprise OS Software*.

Configuring Static Routes

A static route is a user-defined route by which a network can be reached. You can configure as many static routes as desired.

Procedure

To set a static route, use:

```
ADD -IP ROUTe <IP address> [<mask>] {<gateway> | !<port>} <metric>
  [Override]
```

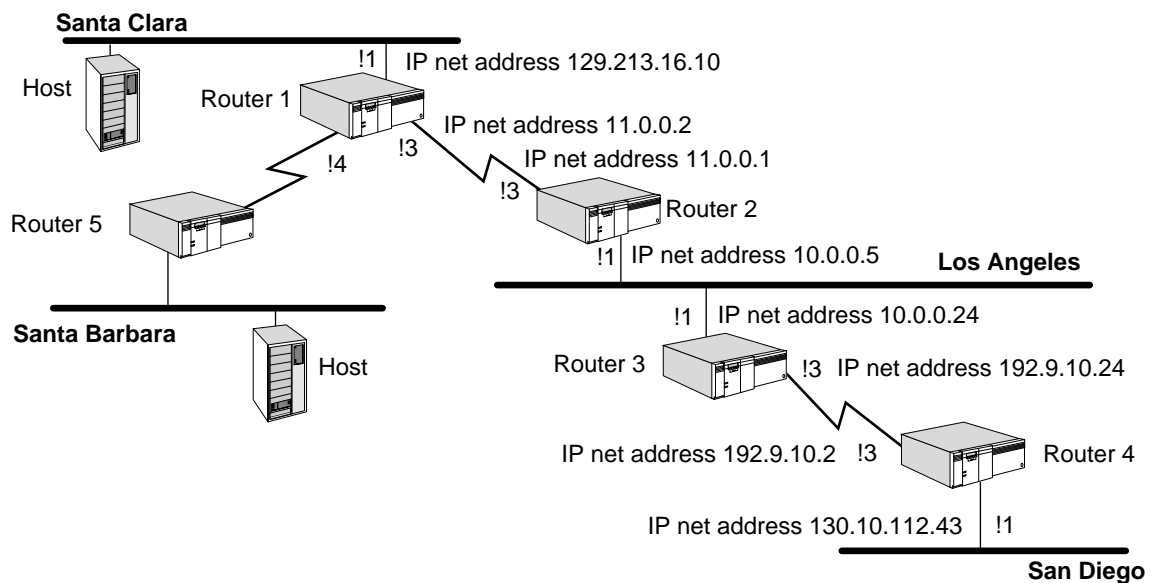
To delete a static route, use:

```
DELeTe -IP ROUTe <IP address> {<gateway> | !<port>}
```

Related Information

The following information pertains to static and dynamic routes.

Figure 54 Routing Between Gateways



See the example in Figure 54. On router 1, you can add a static route for the Los Angeles network by entering:

```
ADD -IP ROUTe 10.0.0.0 11.0.0.1 1
```

This example shows that network number 10.0.0.0 (the Los Angeles network) is reachable through gateway 11.0.0.1. The gateway address is the Internet address of port 3 on router 2. Because a packet routed from router 1 to the Los Angeles network has to go through one gateway, the metric is 1.



The gateway must be located on a network directly connected to the router on which you add the static route. For example, in Figure 55, routers 1 and 2 both have an interface to a common network.

If the outgoing interface is a PPP link (either numbered or unnumbered), you can add a static route using the outgoing port number instead of the next-hop gateway address. For example, on router 1, you can add a static route for the Los Angeles network by entering:

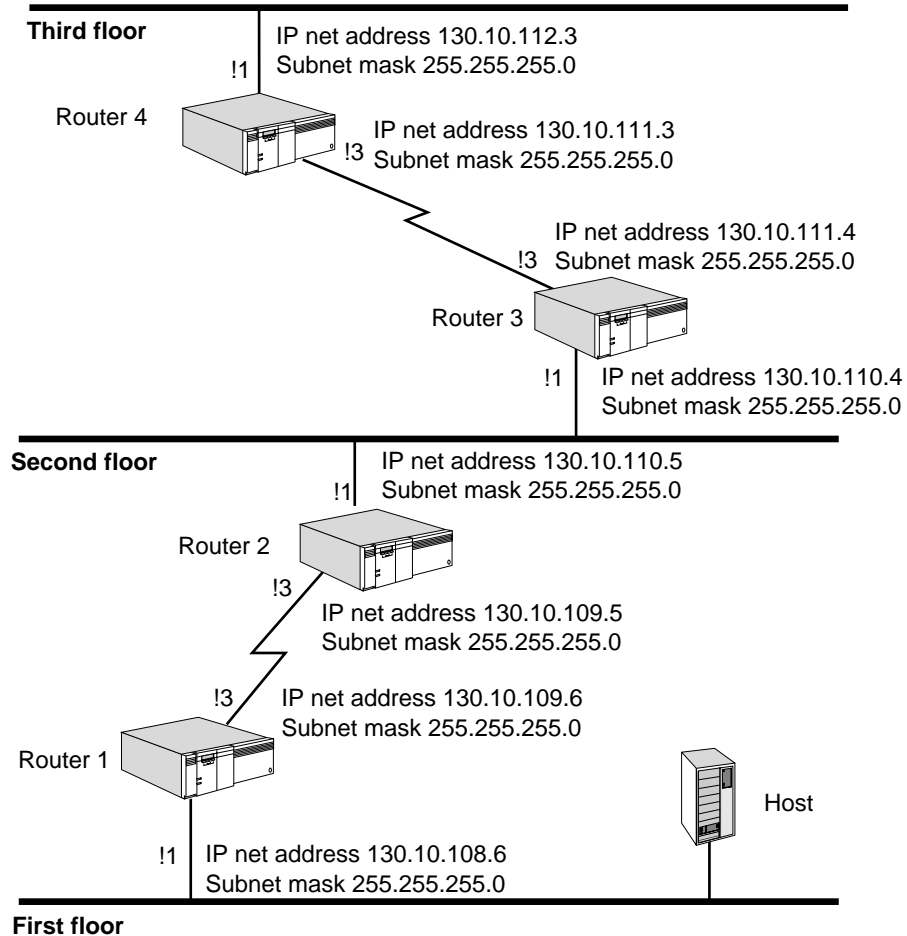
```
ADD -IP ROUTe 10.0.0.0 !3
```

This command achieves the same results as the command in which you entered the gateway address 11.0.0.1 as explained in the previous example.

If the PPP link is unnumbered (no IP network address is configured), you *must* provide the outgoing interface port number because the next-hop gateway address is not available.

Subnet Masks See Figure 55.

Figure 55 Adding a Route Statically in a Subnet Masked Environment



You can also add a route to a subnet in router 1 using a mask by entering:

```
ADD -IP ROUTe 130.10.112.0 255.255.255.0 130.10.109.5 3
```

This command adds the address 130.10.112.0 with subnet mask 255.255.255.0 to the routing table. If a destination network is reachable with both a static route and a learned route, the router uses the static route unless you specify the optional Override value in the ADD ROUTe command. In that case, if a learned route of higher precedence is available, it overrides the static route. (For information on precedence, see "Multipath Routing" later in this chapter). The Override value is entered at the end of the command.

To add the same static route as described earlier with the Override (o) value included, enter:

```
ADD -IP ROUTe 130.10.112.0 255.255.255.0 130.10.109.5 3 Override
```

Configuring Packet Filtering

The IP router supports packet filtering, which controls traffic on your IP network.

Procedure

To configure filters for your IP router, follow these steps:

- 1 Set up a filter policy or policies using:

```
ADD -IP FilterAddrs <adr1> [<dir>] <adr2> [<action> [<protocol>
[<filterID>]]<action> = {PROTOCOLRsrv=<tag>}|
Discard | DODdiscard | Forward | {QPriority = H | M | L} | X25Profile =
<profile>} <protocol> = DLSW | FTP | IP | IPDATA | ICMP | SMTP | TCP |
TELNET | UDP
```

- 2 Create a filter or filters, if required, using:

```
ADD !<filterid> -IP Filters <condition> [,<condition...>] <condition> =
<%offset>:[<operator>]<%pattern>
```

- 3 Set the FilterDefAction parameter using:

```
SETDefault -IP FilterDefAction = [Forward | Discard]
```

- 4 Enable packet filtering by entering:

```
SETDefault -IP CONTROL = Filtering
```

For complete information on the parameters used in this procedure, see the IP Service Parameters chapter in *Reference for Enterprise OS Software*.

Related Information

This section describes the two components of IP packet filtering. It also describes protocol reservation and the PROTOCOLRsrv=<tag> action option.

IP Packet Filtering Components The IP packet filtering feature is composed of two components: setting up a filter policy and creating a filter. To configure this feature, you must set up a filter policy and depending on your filtering needs, you may or may not need to create a filter. If you want to filter packets based on general criteria such as protocol or IP address, you can configure this type of filtering by setting up a filter policy only. If you want to filter packets based on more specific criteria that requires the system to examine the bytes of a packet, you need to set up a filter policy and create a filter.

If you configure a filter policy only, use the protocol field of the ADD -IP FilterAddrs command to specify a protocol you want to filter. If you configure both a filter policy and a filter, use the protocol field to specify the starting point for the offset of a condition. For complete information on the -IP FilterAddrs parameter, see the IP Service Parameters chapter in *Reference for Enterprise OS Software*.

Protocol Reservation One of the action options for the -IP FilterAddrs parameter is PROTOCOLRsrv=<tag>, which is used to set up protocol reservation. Protocol reservation assigns a percentage of bandwidth to designated packets that pass through a specified port and meet certain conditions. The conditions can be protocol type, packet length, packets destined for specified address, and so on.

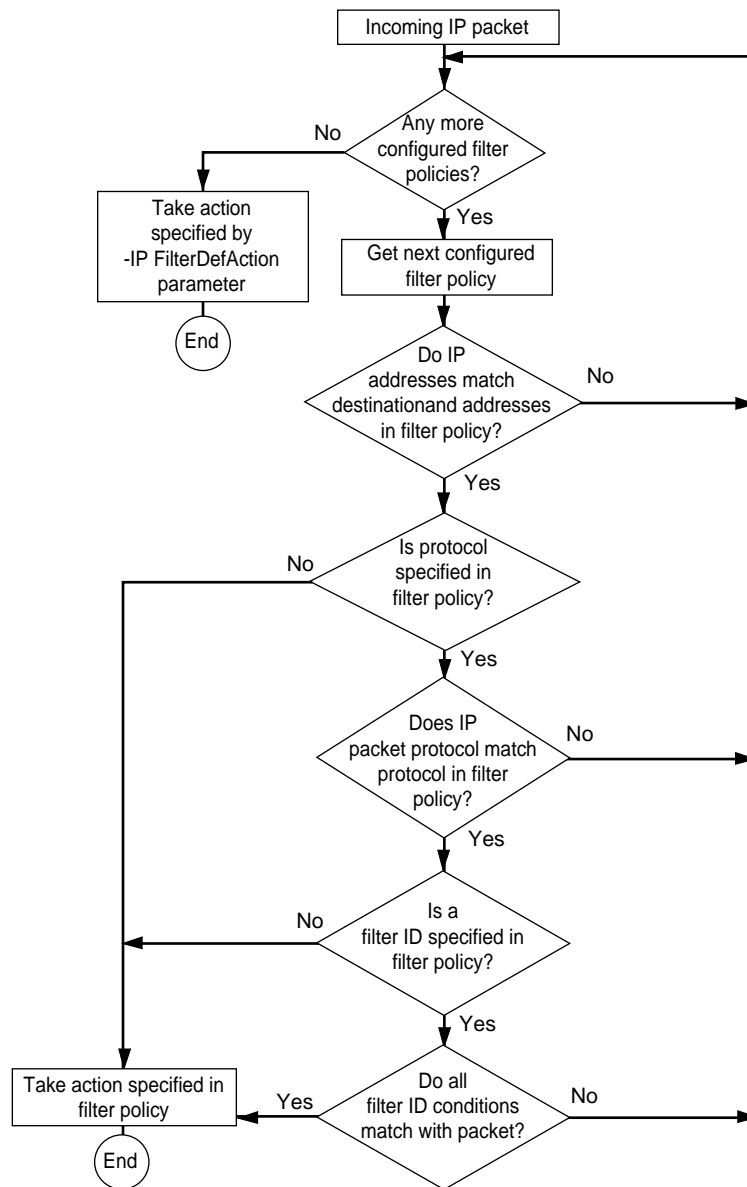
Protocol reservation is set up with different procedures for different packet types. The IP filtering procedure, is applied only to IP-routed packets. IP-routed packets are also filtered using the IP firewall feature. See the Building Internet Firewalls chapter for detailed information about IP firewall.

For a detailed description of the procedures to configure protocol reservation for the different packet types, see the Configuring Protocol Reservation chapter.

For examples of protocol reservation for designated IP-routed packets using the IP filtering procedure and the PROTOcolRsrv=<tag> action option, see “IP Filtering Examples,” [Example 8](#), and [Example 9](#) later in this chapter.

The flowchart in Figure 56 describes how the IP packet filtering feature works. This figure assumes that IP filtering is enabled and at least one filter policy has been configured.

Figure 56 IP Packet Filtering



Enabling the packet filtering feature can have a significant impact on IP router performance. The performance is affected because the verification and

decision-making process that takes place after each packet is received requires significant amounts of processing power.

IP Filtering Examples The following examples show how to configure the IP packet filtering feature.

Example 1 A router with the IP address of 129.213.16.0 and the subnet mask of 255.255.252.0 connects a local company network to the Internet. You want these router operations:

- Allow outgoing Transmission Control Protocol (TCP) connections from hosts on the local network to any host on the Internet.
- Not allow incoming TCP connections except for electronic mail (Simple Mail Transfer Protocol (SMTP), destination port %19) from a host on the Internet to a mail server (129.213.16.9) on the local network.
- Allow Internet Control Message Protocol (ICMP) messages from the Internet for feedback.
- Not allow any other packets to pass through this router.

To establish a filter that allows hosts on the local network to make TCP connections to hosts on the Internet, enter:

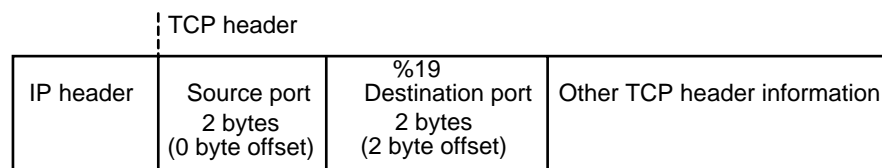
```
ADD -IP FilterAddrs 129.213.16.0/0.0.3.255 > ALL Forward TCP
```

To create a filter policy and filter (filter 1) that allows TCP connections from the Internet only to the local mail server provided that conditions specified in filter 1 are met, enter:

```
ADD -IP FilterAddrs ALL > 129.213.16.9 Forward TCP 1
ADD !1 -IP Filters %2:%0019
```

The ADD -IP FilterAddrs command specified above tells the system to look for TCP packets. The system automatically adjusts for any IP options. Once the system determines that it has received a TCP packet, it looks at the packet's TCP header. The system looks at the 2-byte offset (in the destination port field) for the hexadecimal value %19 (SMTP port) as specified by the ADD -IP Filter command above. Figure 57 shows the TCP header of a packet that meets the criteria established by these commands.

Figure 57 TCP Header of Packet that Meets Filtering Criteria



To establish a policy that allows ICMP messages from the Internet for feedback, enter:

```
ADD -IP FilterAddrs all > 129.213.16.0/0.0.3.255 Forward ICMP
```

To discard all other packets that do not meet any of the criteria discussed previously, enter:

```
SETDefault -IP FilterDefAction = Discard
```

Example 2 You want to set up the following conditions for a host with the address of 129.213.128.1:

- Allow incoming Telnet connections for destination port %17.
- Do not allow incoming TCP connections to the host for well-known ports (ports less than %400).
- Allow all other packets.

To set up a filter (filter 1) and corresponding policy that allows incoming Telnet connections for destination port hexadecimal %17 at a 2 byte offset, enter:

```
ADD !1 -IP Filters %2:%0017
ADD -IP FilterAddrs ALL > 129.213.128.1 Forward TCP 1
```

To set up a filter (filter 2) and corresponding policy that does not allow incoming TCP connections for ports with a value of less than hexadecimal %400 at a 2 byte offset, enter:

```
ADD !2 -IP Filters %2:<%400
ADD -IP FilterAddrs ALL > 129.213.128.1 Discard TCP 2
```

To forward all other packets that do not meet any of the criteria discussed previously, ensure that the -IP FilterDefAction parameter retains its default setting of Forward. If this parameter has been set to Discard, enter:

```
SETDefault -IP FilterDefAction = Forward
```

This example demonstrates that discard and forward filters can be combined. It also highlights the fact that the order filters are configured in is important. As soon as the system finds the first match, it stops searching. In this example, it is important that the system examine and forward packets with destination port %17 (filter 1) before examining and discarding packets with a destination port of less than %400 (filter 2).

Example 3 You want a router to do the following operations:

- Discard all User Datagram Protocol (UDP) packets with a source port of %161 and a destination port of %162.
- Discard all UDP packets if the tenth byte of data has the value of %60.
- Forward all other packets.

To set up a filter (filter 1) and corresponding policy that discards all UDP packets with a source port of hexadecimal %161 at a 0-byte offset and a destination port of hexadecimal %162 at a 2-byte offset, enter:

```
ADD !1 -IP Filters %0:%161, %2:%162
ADD -IP FilterAddrs ALL> ALL Discard UDP 1
```

When creating a filter using the ADD -IP Filters command, separating conditions with a comma (,) as shown in this example, indicates the creation of multiple conditions. If a filter has multiple conditions, all conditions must be satisfied for a match to take place.

To set up a filter (filter 2) and corresponding policy that discard all UDP packets if the tenth byte of data has the value of hexadecimal %60 at an offset of hexadecimal %a, enter:

```
ADD !2 -IP Filters %a:%60
ADD -IP FilterAddrs ALL > ALL Discard UDP 2
```


To forward all other packets that do not meet any of the criteria discussed previously, ensure that the `-IP FilterDefAction` parameter retains its default setting of `Forward`. If this parameter has been set to `Discard`, enter:

```
SETDefault -IP FilterDefAction = Forward
```

Example 4 You want your router to do the following operations:

- Forward all UDP packets from a host with the address 129.213.16.9 except for RPIIP packets (destination port %208).
- Forward all other packets.

To set up a filter (filter 2) and corresponding policy that forwards all UDP packets from host 129.213.16.9 except for RPIIP packets (destination port hexadecimal %208) with a 2-byte offset, enter:

```
ADD !2 -IP Filters %2:%0208  
ADD -IP FilterAddrs 129.213.16.9 > ALL Discard UDP 2
```

To forward all other packets that do not meet any of the criteria discussed previously, ensure that the `-IP FilterDefAction` parameter retains its default setting of `Forward`. If this parameter has been set to `Discard`, enter:

```
SETDefault -IP FilterDefAction = Forward
```

Example 5 To assign a low priority to FTP packets going to and coming from host 129.0.0.2., enter:

```
ADD -IP FilterAddrs ALL < 129.0.0.2 QPriority Low FTP
```

Example 6 To assign 8 as the X.25 profile ID when sending IP traffic over X.25 to host 129.0.0.3, enter:

```
ADD -IP FilterAddrs ALL > 129.0.0.3 X25PROFileid=8
```

The ID of the profile created from the PROFile Service is 8.

Example 7 If the `DodDiscard` action in the `FilterAddrs` parameter is enabled, specified traffic is discarded if a dial-up path is down. If the dial-up path is up, the specified traffic is forwarded.

To mark ICMP traffic from host 10.0.0.1 to host 129.0.0.3 as `DodDiscard`, enter:

```
ADD -IP FilterAddrs 10.0.0.1 > 129.0.0.3 DodDiscard ICMP
```

Example 8 You want to add and set up the following filtering for your bridge/router:

- Add an IP filter that assigns 20 percent of reserved bandwidth for all Telnet sessions, and 30 percent of reserved bandwidth for all FTP packets, sent out through port 2.
- Set the `IP FilterDefAction` parameter so that all packets that do not meet the filtering conditions are forwarded.

To set up these filtering operations, follow these steps:

- 1 Add an IP filter that assigns 20 percent of reserved bandwidth to a PROTOCOLRsrv tag of "Telnet-tag" for all Telnet packets being sent out through port 2, and 30 percent of reserved bandwidth to a PROTOCOLRsrv tag of "FTP-tag" for all FTP packets being sent out through port 2, by entering:

```
ADD -IP FilterAddrs all all PROTOCOLRsrv = Telnet-tag Telnet  
ADD -IP FilterAddrs all all PROTOCOLRsrv = FTP-tag FTP
```

- 2 Add an IP filter default action that forwards any packets that do not satisfy the filter requirements by entering:

```
SETDefault -IP FilterDefAction = Forward
```

- 3 Enable the IP filtering feature by entering:

```
SETDefault -IP CONTROL = Filtering
```

- 4 Assign 20 percent of bandwidth to the PROTOcolRsrv name tag "Telnet-tag" and 30 percent of the bandwidth to the PROTOcolRsrv name tag "FTP-tag" for port 2 by entering:

```
ADD !2 -PORT PROTOcolRsrv Telnet-tag 20
```

```
ADD !2 -PORT PROTOcolRsrv FTP-tag 30
```

- 5 Set PROTOcolRsrv as the option for port 2 by entering:

```
SETDefault !2 -PORT QueueCONTROL = PROTOcolRsrv
```

After you have made these entries, any packet sent out by the system through port 2 that has the name tag "Telnet-tag" will be allocated 20 percent of the bandwidth, and all packets with the name tag "FTP-tag" will be allocated 30 percent of the bandwidth.

Example 9 You want to add and set up the following filtering for your bridge/router:

- Add an IP filter that assigns 10 percent of reserved bandwidth for all FTP packets being sent out to the IP address 50.0.0.1 through port 3.
- Set the IP FilterDefAction parameter so that all packets that do not meet the filtering conditions are forwarded.

To set up these filtering operations, follow these steps:

- 1 Add an IP filter that assigns reserved bandwidth to a PROTOcolRsrv tag of "FTP-tag" for all FTP packets being sent out to the IP address 50.0.0.1 by entering:

```
ADD -IP FilterAddrs all 50.0.0.1 PROTOcolRsrv = FTP-tag FTP
```

- 2 Add an IP filter default action that forwards any packets that do not satisfy the filter requirements by entering:

```
SETDefault -IP FilterDefAction = Forward
```

- 3 Enable the IP filtering by entering:

```
SETDefault -IP CONTROL = Filtering
```

- 4 Assign 10 percent of bandwidth to the PROTOcolRsrv name tag "FTP-tag" for port 3 by entering:

```
ADD !3 -PORT PROTOcolRsrv FTP-tag 10
```

- 5 Set PROTOcolRsrv as the option for port 3 by entering:

```
SETDefault !3 -PORT QueueCONTROL = PROTOcolRsrv
```

After you have made these entries, any packet forwarded by the system to port 3 that has the name tag "FTP-tag" will be allocated 10 percent of the bandwidth.

Configuring RIP Routing Policies

The routing policies supported by RIP allow you to control the reporting of routing information on a per-port basis. This section describes the various routing policies

you can configure and the parameters associated with configuring each policy, and provides examples of configuring policies.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Familiarize yourself with the various policies that are available, then determine which policies you want to configure. Table 12 lists and briefly describes each policy and its associated parameter.
- If you plan to access or receive information on routes from specific networks as opposed to all or no networks, determine the IP addresses of these specific networks.

Table 12 RIP Routing Policies

Policy	Description	Parameter
Advertise	Controls which routes are reported regardless of the route source.	AdvertisePolicy
Static	Controls which static routes are reported in the IP routing environment.	StaticPolicy
Exterior	Controls which BGP is reported.	ExteriorPolicy
Interior	Controls which OSPF or IISIS routes are reported.	InteriorPolicy
Receive	Controls which RIP routes are received by a trusted neighbor.	ReceivePolicy

For more information on the parameters listed in this table, see the RIPIP Service Parameters chapter in *Reference for Enterprise OS Software*.

Procedure

To configure a routing policy, follow these steps:

- 1 Establish an advertise policy that controls the advertisement of routes through RIP regardless of the source from which the route is learned. Use:

```
ADD !<port> -RIPIP AdvertisePolicy All | None | [~]<IP address> [<metric>
(0-15)]
```

For example, to configure a policy on port 1 that forwards information on all routes to network 10.0.0.0, enter:

```
ADD !1 -RIPIP AdvertisePolicy 10.0.0.0
```

In this example, a metric associated with network 10.0.0.0 was not specified. If you decide not to specify a metric with the AdvertisePolicy parameter or to specify a metric of zero, a route is reported with a metric calculated from the routing table.

- 2 Establish a receive policy that accepts or refuses to accept information on routes learned by RIP from a trusted neighbor. Use:

```
ADD !<port> -RIPIP ReceivePolicy All | None | [~]<IP address> [<metric>
(0-15)]
```

For example, to configure port 1 so that it accepts information on routes learned by RIP for network 10.0.0.0, enter:

```
ADD !1 -RIPIP ReceivePolicy 10.0.0.0
```

In this example, a metric associated with network 10.0.0.0 was not specified. If you decide not to specify a metric with the ReceivePolicy parameter or specify a metric of zero, a route with the originally reported metric is stored in the routing table.

- 3 To control the reporting of routes learned from specific sources, establish the following policies:

- Exterior policy for routes learned from BGP
- Interior policy for routes learned from OSPF or IISIS
- Static policy for reporting static (user-) configured routes

Use the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters to complete this step. The syntax for the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters is the same as the syntax for the AdvertisePolicy parameter. For the AdvertisePolicy parameter syntax, see step 1.

For example, to configure a policy on port 1 that forwards routing information learned from BGP, OSPF or IISIS, and about static routes configured on network 10.0.0.0, enter:

```
ADD !1 -RIPIP ExteriorPolicy 10.0.0.0
ADD !1 -RIPIP InteriorPolicy 10.0.0.0
ADD !1 -RIPIP StaticPolicy 10.0.0.0
```

In this example, a metric associated with network 10.0.0.0 was not specified in each of the commands. If you decide not to specify a metric with the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters or specify a metric of zero, a route is reported with a metric calculated from the routing table.

In RIPv1, the recipient of a RIP message has no way to distinguish whether a route is within the RIP domain or is imported from an exterior gateway protocol domain (for example, BGP) or another interior gateway protocol domain (for example, OSPF). The inclusion of route tag in RIPv2 allows this information to be propagated along with every route entry. The tag information is always preserved and accompanies the route when the route entry makes its way throughout the RIPv2 domain. This information can be useful for a router when it wants to export the route to another domain, by comparing the route tag (hence the type of route) against its export policy.

For example, to add a route tag to use with this route entry, enter:

```
ADD !1 -RIPIP ExteriorPolicy 10.0.0.0 tag 65
ADD !1 -RIPIP InteriorPolicy 10.0.0.0 tag 65
```

If you decide to have routes reported with a metric calculated from the routing table, you can manipulate the conversion formula that RIP uses to convert a metric from the routing table into one that it understands. To manipulate the formula, go on to step 4; otherwise, you have finished configuring RIP routing policies.

The metric that you configure with the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters is static or unchanging. This is in contrast to the metric that is calculated from the routing table. You can additionally manipulate the formula that is used to calculate the metric. For these reasons, 3Com recommends using the metric that is calculated from the routing table.

- 4 If you configured the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters and want to manipulate the formula that is used to calculate the metric, use:

```
ADD -RIPIP ImportMetric <from protocol> Multiply | Divide <operand>
```

For example, to manipulate the conversion formula used to report OSPF routes so that the metrics reported with these routes are imported into RIP without being changed, enter:

```
ADD -RIPIP ImportMetric OSPF Divide 1
```

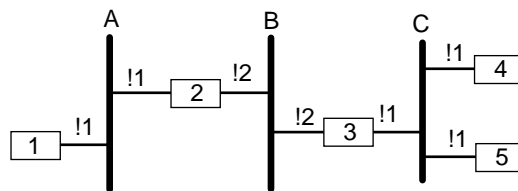
To manipulate the conversion formula used to report OSPF routes so that the metrics reported with these routes are divided by 16, enter:

```
ADD -RIPIP ImportMetric OSPF Divide 16
```

Migration to a RIPV2 Network

In Figure 58, initially all routers are RIPV1 routers.

Figure 58 Migration to a RIPV2 Network



In the first upgrade, routers 2 and 3 are upgraded to 11.0 and are RIPV2 capable. Since network B is now directly connected to the RIPV2 capable routers, RIPV2 capability can be activated on those routers at the interfaces that are directly connected to network B.

```
SETD !2 -RIPIP V2CompatMode = RIPV2
```

Interface 1 of both routers remains at RIPV1 mode (default value).

In the next upgrade, routers 1 and 4 are upgraded to 11.0. Network a now has both 11.0 routers. To activate RIPV2:

```
SETD !1 -RIPIP V2CompatMode = RIPV2
```

Network C still has an old router (router 5). To maintain RIP updates among those routers, do not set the new routers to RIPV2 mode yet. To take advantage of RIPV2 while maintaining the update exchange with the old router, use the RIPV1Compatible mode for the new routers. For routers 3 and 4, enter:

```
SETD !1 -RIPIP V2CompatMode = RIPV1Compatible
```

In the final upgrade, router 5 is upgraded to 11.0. Now you can apply RIPV2 on router 5. Enter:

```
SETD !1 -RIPIP V2CompatMode = RIPV2
```

For complete information on the commands and parameters discussed in this section, see the RIPV2 Service Parameters chapter in *Reference for Enterprise OS Software*.

Configuring OSPF Routing Policies

The routing policies supported by OSPF allow you to control the reporting of routes learned from other sources. This section describes the various routing policies you can configure and the parameters associated with configuring each policy, and provides examples of configuring policies.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Determine which policies you want to configure. Table 13 lists and briefly describes each policy and its associated parameter.
- If you plan to access or receive information on specific routes as opposed to all or no routes, determine the IP addresses of these specific routes.

Table 13 OSPF Routing Policies

Policy	Description	Parameter
Exterior	Controls whether exterior routing protocol (BGP) learned routes are further advertised into the OSPF domain.	ExteriorPolicy
Interior	Controls whether interior routing protocol (RIP or IISIS) learned routes are further advertised into the OSPF domain.	InteriorPolicy
Static	Controls which static routes are further advertised into the OSPF domain.	StaticPolicy
Direct	Controls whether a locally attached network (with OSPF disabled on such interface) should be further advertised into the OSPF domain.	DirectPolicy

For more information on the parameters listed in this table, see the OSPF Service Parameters chapter in *Reference for Enterprise OS Software*.

Procedure

Assume your network topology is similar to that shown in Figure 59. Use the following procedure to control the reporting of routes learned from other sources and advertised into the OSPF domain.

To control the reporting of routes learned from other sources and advertised into the OSPF domain, follow these steps:

- 1 Enable the OSPF Protocol on the appropriate ports on the backbone routers.

For example, on routers 1, 2, and 3, enter:

```
SETDefault #1 -OSPF CONTROL = Enable
SETDefault #2 -OSPF CONTROL = Enable
```

- 2 Configure the backbone routers to learn routes from other interior routing protocols (such as RIPv2) within the same autonomous system.



The default setting of the -OSPF InteriorPolicy parameter is None. This means that if a router runs both the OSPF and RIPv2 Protocols, the routes learned by one of these protocols are not reported to the other.

For example, to configure Router 1 to learn routes from RIPv2 domain #1, on Router 1, enter:

```
ADD -OSPF InteriorPolicy All
```

You could also specify an IP address of the network in RIPv2 domain #1 using:

```
ADD -OSPF InteriorPolicy <IPaddress>
```

- 3 Configure the backbone routers to learn routes from other exterior routing protocols, such as BGP, in another autonomous system.

For example, to configure router 3 to learn routes from autonomous system 2, on router 3, enter:

```
ADD -OSPF ExteriorPolicy All
```

- 4 In a Boundary Routing environment, configure the backbone router to advertise routes from the remote domain into the OSPF domain.

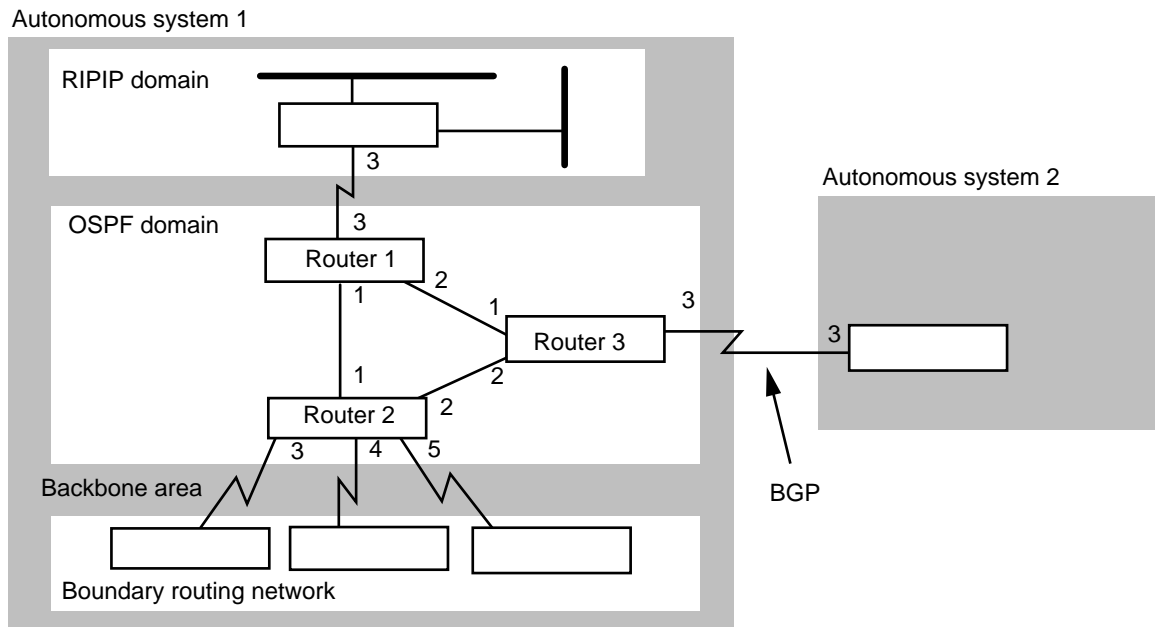
For example, on router 2, OSPF is disabled on wide area ports 3, 4, and 5. In order for router 2 to advertise these routes, on router 2, enter:

```
SETDefault !3 -OSPF DirectPolicy = Advertise
```

Enter the same command for ports 4 and 5.

The DirectPolicy parameter applies to directly attached networks and only applies to ports where the -OSPF CONTROL parameter is set to Disable.

Figure 59 OSPF Routing Policies



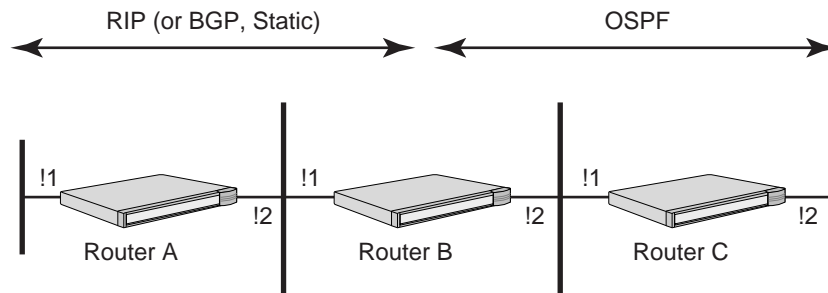
Configuring OSPF Router Aggregation

In OSPF, you can import routes from an exterior routing protocols such as BGP, an interior routing protocol, such as RIP, static routes, and directly connected networks. These imported routes become OSPF external routes. In some networks there are too many OSPF external routes.

OSPF router aggregation lets you define external route ranges to aggregate the area's routes, which reduces the routing information in the backbone and subsequently in all areas. If an external route is within the range and the range is configured as "advertise," then the range network is advertised. If several external routes are in the same external route range, then only one network is advertised. This reduces the number of routes in the backbone and regular areas.

To establish OSPF router aggregation use:

```
ADD -OSPF ExtRouteraNges <IP address><mask> [Tag] [Advertise | DontAdvertise]
```

Figure 60 OSPF Router Aggregation

In the example shown in Figure 60 RIP is enabled on bridge/router A and !1 on bridge/router B. OSPF is enabled on !2 of bridge/router B and bridge/router C.

To enable OSPF route aggregation, follow these steps:

- 1 On bridge/router B, enter:
ADD -OSPF InteriorPolicy ALL 1
- 2 On bridge/routers B and C, view the external LSAs by entering:
SHoW -OSPF lsd external
Two external LSA are shown in the resulting display.
- 3 On bridge/router B, enter:
ADD -OSPF ERN 10.0.
- 4 To establish OSPF route aggregation on bridge/router B, enter:
ADD -OSPF ExtRouteraNges 10.0.0.0 255.0.0.0

Now, when you enter the SHoW -OSPF LSD external command on bridge/routers C and D, only one LSA appears on the resulting display.

Configuring ISIS Routing Policies

The routing policies supported by ISIS allow you to control the reporting of routes learned from other sources. This section describes the various routing policies you can configure, the parameters associated with configuring each policy, and examples of configuring policies.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Determine which policies you want to configure. Table 14 lists and briefly describes each policy and its associated parameter.
- If you plan to access or receive information on specific routes as opposed to all or no routes, determine the IP addresses of these specific routes.

Table 14 ISIS Routing Policies

Policy	Description	Parameter
Exterior	Controls whether exterior routing protocol (BGP) learned routes are further advertised into the ISIS domain.	ExteriorPolicy

Table 14 ISIS Routing Policies (continued)

Policy	Description	Parameter
Interior	Controls whether interior routing protocol (RIP or OSPF) learned routes are advertised into the ISIS domain.	InteriorPolicy
Static	Controls which static routes are advertised into the ISIS domain.	StaticPolicy

For more information on the parameters listed in this table, see the ISIS Service Parameters chapter in *Reference for Enterprise OS Software*.

Procedure

Assume your network topology is similar to that shown in Figure 61.

To control the reporting of routes learned from other sources and advertised into the ISIS domain, follow these steps:

- 1 Enable the ISIS Protocol on the backbone routers.

For example on routers 1, 2, and 3, enter:

```
SETDefault -ISIS CONTROL = Enable
```

- 2 Configure the backbone routers to learn routes from other interior routing protocols (such as RIP or OSPF) within the same autonomous system.



The default setting of the -ISIS InteriorPolicy parameter is None. This means that if a router runs both the ISIS and RIP Protocols, the routes learned by one of these protocols are not reported to the other.

For example, to configure router 1 to learn routes from RIP domain, on router 1, enter:

```
ADD -ISIS InteriorPolicy All
```

You can specify an IP address of the network in RIP domain using:

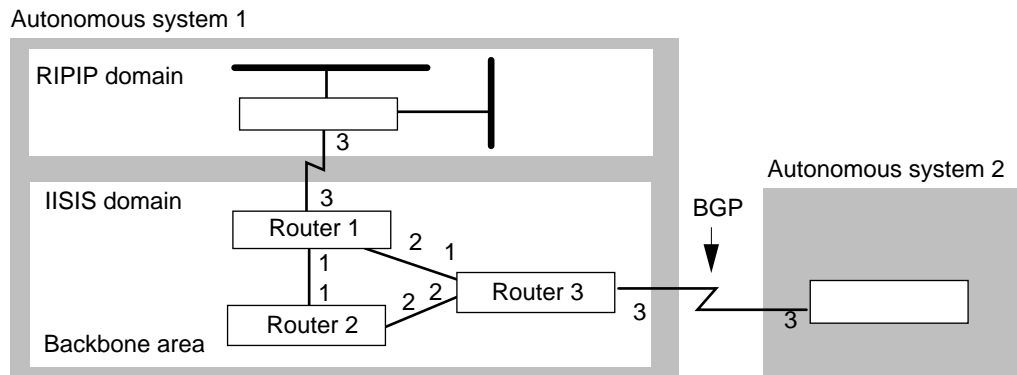
```
ADD -ISIS InteriorPolicy <IPaddress>
```

- 3 Configure the backbone routers to learn routes from other exterior routing protocols, such as BGP, in another autonomous system.

For example, to configure router 3 to learn routes from autonomous system 2, enter the following command on router 3:

```
ADD -ISIS ExteriorPolicy All
```

Figure 61 ISIS Routing Policies



Using the IP Security Option

For more information on using the IP security option, see the IP Security Options chapter.

Configuring Inter-autonomous System Routing Using BGP

This section describes how to configure BGP as your inter-autonomous system routing protocol. You will need to configure the following items:

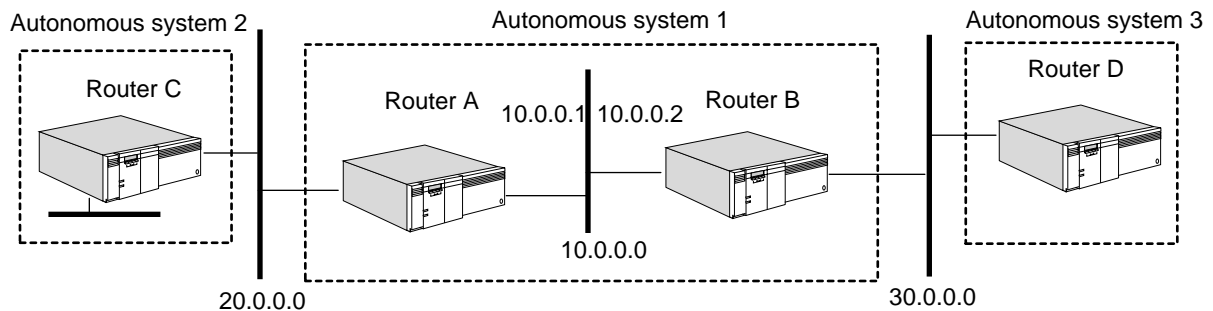
- BGP peers
- Default route
- Route aggregation
- Route importing from an IGP to BGP domain (interior policy)
- Route importing from a BGP to IGP domain (exterior policy)
- Network number policies
- AS-path policies (permit, deny, and weight)

For conceptual information on inter-autonomous system routing, see “Autonomous System Routing Using BGP” later in this chapter.

Configuring BGP Peers

For BGP to learn routes between autonomous systems (ASs) and determine the reachability of networks outside of its AS, you must configure BGP peers. For information on peers, see “External and Internal Peers” later in this chapter.

Figure 62 BGP Peers



To configure router A and router B in Figure 62 as peers, follow these steps:

- 1 Define the local AS number for the routers using:

```
SETDefault -BGP LocalAS = <AS Number>(1-65536)
```

In this example, routers A and B are internal peers and part of AS 1; on each router, enter:

```
SETDefault -BGP LocalAS = 1
```

This parameter defines the AS number used by this BGP speaker in the OPEN message and in all routing updates as the originating AS number. The local AS number also determines whether a peer is connected through an internal or external BGP session.

- 2 Add a peer to each router using:

```
ADD -BGP PEER <IP address> <AS Number> [RouteReflectorClient]
```

AS numbers range from 1 to 65535.

On router B, specify router A's IP address (10.0.0.1) and AS Number (1):

```
ADD -BGP PEER 10.0.0.1 1
```

On router A, specify router B's IP address (10.0.0.2) and AS Number (1):

```
ADD -BGP PEER 10.0.0.2 1
```



The router must know the AS number for itself and the peer that is being added before it can establish a BGP session with its peer.

- 3 Enable BGP routing by entering the following command on both routers:

```
SETDefault -BGP CONTROL = Enable
```

- 4 Enable each peer that you added with the ADD -BGP PEER command using:

```
SETDefault [!<IP address>] -BGP PeerControl = Enable
```

On router B, specify router A's IP address for <IP address>. For example, enter:

```
SETDefault !10.0.0.1 -BGP PeerControl = Enable
```

On router A, specify router B's IP address for <IP address>. For example, enter:

```
SETDefault !10.0.0.2 -BGP PeerControl = Enable
```

Router A establishes a TCP connection with the router B (peer-to-peer communication). After the connection is established, both peers exchange BGP update packets indicating the networks each peer can reach.

- 5 Display routes learned through BGP by entering the following command on any BGP router:

```
SHOW -BGP ROUTE
```

For information on how to read the display, see "ROUTE" in the BGP Service Parameters chapter in *Reference for Enterprise OS Software*.

- 6 Display peer information by entering the following command on any BGP router:

```
SHOW -BGP PEER
```

The display shows the current mapping of peer ID to IP address to AS number and shows the current state of the peer (disabled, open, connecting).

Configuring a Default Route

You can configure a default route in the BGP Routing Table to provide the IP address of a network that can be used as the default network to destinations that are not explicitly listed in the routing table. Configuring a default route is helpful under the following circumstances:

- The routing policy of a peer does not permit the advertisement of a default route.
- When the local router is unable to maintain the complete BGP Routing Table due to memory limitations.

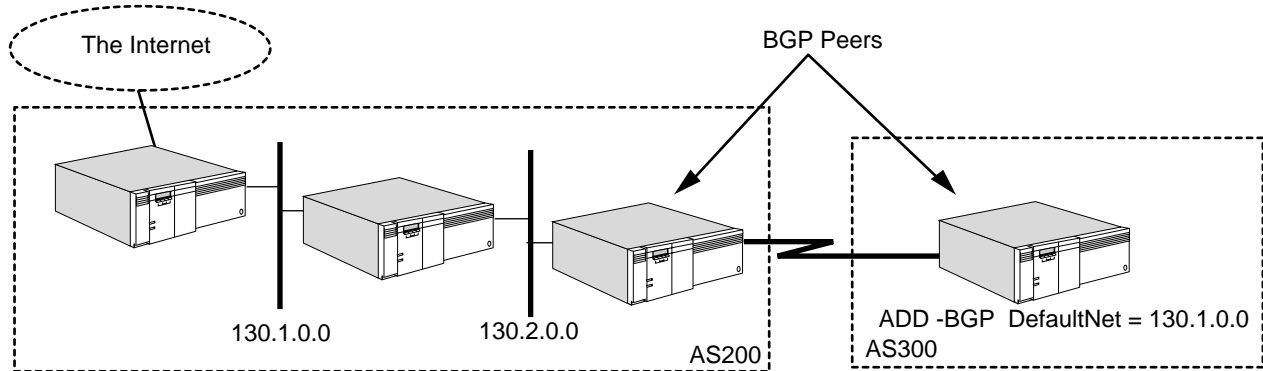
If a route for a particular destination address is not contained in the BGP Routing Table, BGP checks for a default route. To configure a default route, see Figure 63 and use:

```
ADD -BGP DefaultNet <IP address>
```

The configured IP address does not have to be a directly connected network. As long as the local router has a route to the IP address, it can forward all default

route traffic to the IP address. The next-hop address in the BGP Routing Table is automatically calculated by the system software.

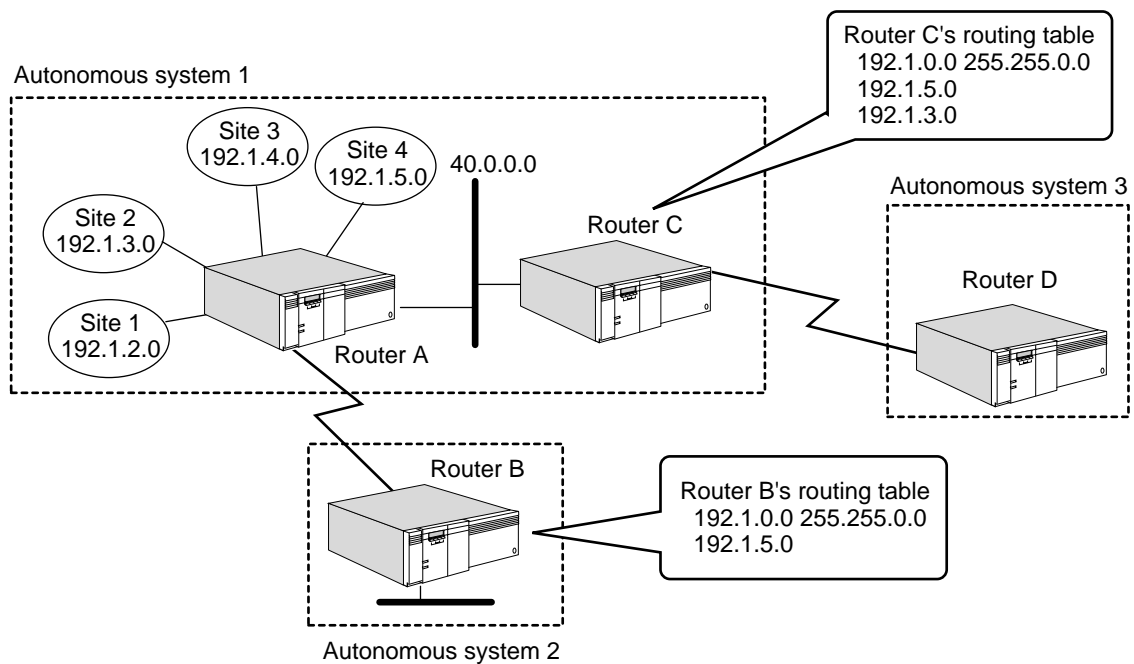
Figure 63 BGP Default Route



Configuring BGP Route Aggregation

BGP route aggregation uses the Classless InterDomain Routing (CIDR) address aggregation strategy to combine the characteristics of several different routes so that a single route can be advertised (see Figure 64). By combining several networks into one supernet, the number of BGP messages sent to peers and the size of the routing table are reduced. Unnecessary details about subnets are hidden from peers.

Figure 64 BGP Route Aggregation



Before beginning the procedure, make sure you have completed “Configuring BGP Peers” earlier in this chapter (making routers A and B peers, and A and C peers), referring to Figure 62.

To configure BGP route aggregation, see Figure 64 and follow these steps:

- 1 Specify a list of networks that BGP advertises as a single supernet route by using:

```
ADD -BGP AggregateRange <IP address> <mask>
```

For example on router A, combine the routes to sites 1, 2, and 3 into a range so that only a single route is advertised. Enter:

```
ADD -BGP AggregateRange 192.1.0.0 255.255.0.0
```



Aggregation should never enclose Class D address space (224.0.0.0 through 239.255.255.255).

- 2 Specify a list of routes that BGP explicitly advertises using:

```
ADD -BGP AggregateExcept <IP address> <mask>
```

For example, if you do not want site 4 included in the aggregation range, enter:

```
ADD -BGP AggregateExcept 192.1.5.0 255.255.255.0
```

- 3 Enable route aggregation and the BGP routing protocol by entering:

```
SETDefault -BGP CONTROL = (Enable, AGgregate)
```

As shown in the Figure 64, router A can advertise a single network (192.1.0.0/255.255.0.0) that summarizes each of the three connected sites and also explicitly advertises the exception route (192.1.5.0). Without the use of CIDR, router A advertises each route with a separate entry, and router B's routing table grows in size. With route aggregation, router B's routing table has an entry for 192.1.0.0 and 192.1.5.0.

Explicit routes within an aggregate can be advertised by the following optional configuration. If 192.1.3.0 should be explicitly advertised to all internal peers, follow these steps:

Add a network filter (network address and mask) using:

```
ADD -BGP NetworkFilter <NetfilterID> <network address> <mask>
```

For example on router A, configure route 192.1.3.0 as network filter 1 by entering:

```
ADD -BGP NetworkFilter 1 192.1.3.0 255.255.255.0
```

- 4 Apply a network policy for internal peers to advertise outgoing routes using:

```
ADD -BGP NetPolicyInt <NetfilterID> Explicit
```

For example on router A, associate network filter 1 (192.1.3.0 255.255.255.0) to be explicitly advertised to router C by entering:

```
ADD -BGP NetPolicyInt 1 Explicit
```

The router C routing table has entries for 192.1.0.0, 192.1.5.0, and 192.1.3.0.

The NetPolicyAll, NetPolicyExt, NetPolicyPeer parameters can also be configured for explicit policies that are applied to outgoing constituent routes of aggregates. For more information, see "Route Aggregation" later in this chapter, and the BGP Service Parameters chapter in *Reference for Enterprise OS Software*.

Importing Routes from IGP to a BGP Domain

To control how route reachability information is shared between routers in different domains, the BGP router can be configured to accept or reject Interior Gateway Protocol (IGP) routing information. You can configure an interior policy on your BGP router to control the import (also known as *route leaking*) and

advertisement of routes from an IGP domain. IGP refers to protocols (such as RIP, OSPF, and IISIS) that operate within a domain.

You must control how you import routes from an IGP domain to BGP domain when your network is connected to the Internet. 3Com recommends that you statically map valid routes and have only these routes imported into the BGP domain. Otherwise, dynamically changing routes in the IGP domain are constantly imported into the BGP domain causing increased load on all core routers to process unnecessary route flaps (routes coming up and going down).

To import a route from an IGP domain into a BGP domain, follow these steps:

- 1 Define the network filters specifying the network address and mask using:

```
ADD -BGP NetworkFilter <NetfilterID> <network address> <mask>
```

For example, to identify networks that have a value of 11.5.7 in the first three octets, enter:

```
ADD -BGP NetworkFilter 2 11.5.7.0 255.255.255.0
```

To identify networks that have a value of 193.4 in the first two octets, enter:

```
ADD -BGP NetworkFilter 5 193.4.0.0 255.255.0.0
```

To identify networks that have a value of 193.7.8 in the first three octets, enter:

```
ADD -BGP NetworkFilter 6 193.7.8.0 255.255.255.0
```

- 2 Advertise only the specified networks and block all others, or block only the specified networks and advertise all others using:

```
ADD -BGP InteriorPolicy <NetfilterID> <Permit | Deny>
```

By default, no IGP route (including static routes and directly connected routes) are imported into the BGP routing table. To set up whether IGP routes are imported into the BGP routing table, use the SETDefault -BGP IntPolDefault command. For more information about this parameter, see the BGP Service Parameters chapter in *Reference for Enterprise OS Software*.

- For example, to import only networks that have a value of 11.5.7 in the first three octets (filter 2), all networks that have a value of 193.4 in the first two octet (filter 5), and all networks that have a value of 193.7.8 in the first three octets (filter 6), enter:

```
ADD -BGP InteriorPolicy 2 Permit
```

```
ADD -BGP InteriorPolicy 5 Permit
```

```
ADD -BGP InteriorPolicy 6 Permit
```

- For example, to block the import of only the specified networks and import all others (except OSPF routes), enter:

```
ADD -BGP InteriorPolicy 2 Deny
```

```
ADD -BGP InteriorPolicy 5 Deny
```

```
ADD -BGP InteriorPolicy 6 Deny
```

To block the import of OSPF Type 1 external routes on the specified networks, enter:

```
ADD -BGP OspfExtPolicy 2 Deny ExType1
```

```
ADD -BGP OspfExtPolicy 5 Deny ExType1
```

```
ADD -BGP OspfExtPolicy 6 Deny ExType1
```



To avoid an invalid configuration and the interior policy from being ignored, do not configure the `InteriorPolicy` or `OspfExtPolicy` parameters with a mixture of permit and deny policies. You must specify the policy as either all permit or all deny policies.

- Restart all BGP sessions to recompute the route selection process by entering:

```
SETDefault -BGP CONTROL = Enable
```

For more information, see “Interior Policies” later in this chapter.

Importing Routes from a BGP Domain to an IGP Domain

To control how route reachability information is shared between routers in different domains (BGP to IGP), the router can be configured to accept or reject injection of BGP routing information into the IGP routing domain. For stub and multi-homed ASs, you can configure a default route on your IGP router. For transit ASs, you can configure an exterior policy to control the import (also known as route leaking) and advertisement of routes from BGP, or just run an IGP between ASBRs. In most cases, importing routes is probably not required.

Stub Autonomous Systems A stub AS has only one connection to another AS.

In a stub AS using RIPV1, RIP cannot advertise both the network number and the mask. As a result, some BGP-derived routes may not be understood by RIP. You need to configure the `-RIP DefaultMetric` parameter on the border router to advertise a default route (to network 0.0.0.0) using:

```
SETDefault !<port> -RIP DefaultMetric = <metric> (0-15)
```

In a stub AS using OSPF or IISIS, these protocols can advertise both the network number and the mask, so you can import BGP-derived routes and advertise them by OSPF or IISIS. However, the simplest solution is to configure the OSPF or IISIS `DefaultMetric` parameter on the Autonomous System Boundary Router (ASBR) to generate an external link state advertisement (LSA) for network 0.0.0.0 using:

```
SETDefault -OSPF DefaultMetric = [Disable | <metric>(1-65535) [Type1 | Type2]]
SETDefault -IISIS DefaultMetric = Disable | <metric> (1-63) [Internal | External]
```

Multi-homed Autonomous Systems A multi-homed AS has connections to more than one AS but does not carry transit traffic. All of the traffic in a multi-homed AS is considered local.

In a multi-homed AS executing RIP, some BGP-derived routes may not be understood by RIP; the `-RIP DefaultMetric` parameter should be configured on *each* border router to advertise the default route. The default route allows routers that are internal to the AS to select the least-cost default route to forward traffic destined for another AS.

In a multi-homed AS executing OSPF or IISIS, all BGP-derived routes can be advertised by OSPF, but it is not always necessary or desirable to import them into a multi-homed autonomous system. The easiest solution is to configure the OSPF or IISIS `DefaultMetric` parameter to `Type1` on *each* ASBR to generate an external LSA for network 0.0.0.0. When each internal router constructs its shortest path tree, the router selects the least cost default route.

Transit Autonomous Systems A transit AS has connections to more than one AS and carries both local traffic and transit traffic. Transit traffic is any traffic that does not originate or terminate within the local AS.

Every router within a transit AS must have explicit routing information for all networks that make up the internetwork. Each internal router must be able to forward a packet to any destination without relying on a default route. As a result, RIP cannot be used as the IGP for a transit AS. A transit AS requires a protocol that scales and supports the advertisement of BGP-derived routes. A transit AS must use BGP on all border routers, and OSPF, ISIS, or other protocols capable of conveying network masks. An IGP is the best choice, but may not scale well.

To allow the import of BGP routes into the IGP, use:

```
ADD -OSPF ExteriorPolicy All | None | [~]<IP address> <metric> [Type1 |
    Type2]
ADD -ISIS ExteriorPolicy All | None | [~]<IP address> <metric> [Internal |
    External]
```

For more information, see “Exterior Policies” later in this chapter.

Configuring Network Number Policies

You can control the receipt (import) or advertisement (export) of BGP routes based on the network address using permit or deny network policies. Permit or deny policies are applied to incoming or outgoing packets, or to both incoming and outgoing packets.

To configure a network policy, follow these steps:

- 1 Identify the individual network number or block of network numbers to which the policy will be applied using:

```
ADD -BGP NetworkFilter <NetfilterID> <network address> <mask>
```

For example, to configure network filter 1 for network addresses starting with 192.2.1, enter:

```
ADD -BGP NetworkFilter 1 192.2.1.0 255.255.255.0
```

- 2 Define the policy: which peer the policy applies to, the type of policy (permit or deny) and whether the policy applies to incoming packets, outgoing packets, or both.

The following syntaxes can be used:

```
ADD -BGP NetPolicyAll <NetfilterID> {Permit | Deny [In | Out | Both]} |
    Explicit
ADD -BGP NetPolicyExt <NetfilterID> {Permit | Deny [In | Out | Both]} |
    Explicit
ADD -BGP NetPolicyInt <NetfilterID> {Permit | Deny [In | Out | Both]} |
    Explicit
ADD [!<IP address>] -BGP NetPolicyPeer <NetfilterID> {Permit | Deny [In |
    Out | Both]} | Explicit
```

For example, to permit the import of routes to 192.2.1.0 from all peers, enter:

```
ADD -BGP NetPolicyAll 1 Permit In
```

To permit the import of routes to 192.2.1.0 from external peers, enter:

```
ADD -BGP NetPolicyExt 1 Permit In
```

To permit the import of routes to 192.2.1.0 from internal peers, enter:


```
ADD -BGP NetPolicyInt 1 Permit In
```

To permit the import of routes to 192.2.1.0 to be accepted from peer 10.0.0.2, enter:

```
ADD -BGP !10.0.0.2 NetPolicyPeer 1 Permit In
```

If this is the only policy defined, all other routes from peer 10.0.0.2 are discarded. You can configure deny policies using the same syntaxes by specifying Deny instead of Permit.



All policies in a specific direction (in/out) must be either all permit policies or all deny policies. A mix of permit and deny policies causes ambiguity resulting in the entire policy list being ignored.

- Restart all BGP sessions to recompute the route selection process by entering:

```
SETDefault -BGP CONTROL = Enable
```

For more information, see “Network Number-Based Policies” later in this chapter.

Configuring AS-Path Permit or Deny Policies

You can control the receipt (import) or advertisement (export) of BGP routes based on presence or absence of specific AS numbers in the AS-PATH attribute. Recall that the AS-PATH attribute is contained in each update message. For information about this attribute, see “Path Attributes” later in this chapter.

Permit or deny policies are applied to incoming packets, outgoing packets, or to both inbound and outbound packets, and help filter incoming and outgoing routes.

To configure a AS-path permit or deny policy, follow these steps:

- Define the filter to which the policy will apply using:

```
ADD -BGP AsFilter <AsfilterID> "<regular expression>"
```

For examples of regular expressions, see “Regular Expressions Examples” later in this chapter.

- Define the policy: which peer the policy applies to, the type of policy (permit or deny) and whether the policy applies to incoming packets, outgoing packets, or both.

The following syntaxes can be used to assign policies to all peers, external peers, internal peers, or a specific peer:

```
ADD -BGP AsPolicyAll <AsfilterID> [[Permit | Deny [In | Out | Both]]  
[Weight <weight>]
```

```
ADD -BGP AsPolicyExt <AsfilterID> [[Permit | Deny [In | Out | Both]]  
[Weight <weight>]
```

```
ADD -BGP AsPolicyInt <AsfilterID> [[Permit | Deny [In | Out | Both]]  
[Weight <weight>]
```

```
ADD [!<IP address>] -BGP AsPolicyPeer <AsfilterID> [[Permit | Deny [In |  
Out | Both]] [Weight <weight>]
```



To maintain consistent routing information within an AS, do not apply permit or deny policies to internal BGP peers. The only type of policy that should be applied to internal peers is one that changes the preference by adding additional weight to selected paths.

- Restart all BGP sessions to recompute the route selection process by entering:

SETDefault -BGP CONTROL = Enable

For examples of deny and permit filters, see “Deny Filters Examples” and “Permit Filters Examples”.

Regular Expressions Examples This section shows examples of regular expressions.

Blank spaces are represented in the examples as underscores (_). When two spaces are shown together, a space has been inserted between the underscores, for example _ _. You must enter a blank space for each underscore shown in the examples.

Example 1 To create filter 1 that identifies an AS-PATH attribute containing AS 25, enter:

```
ADD -BGP AsFilter 1 "_25_"
```

Example 2 To create filter 2 that identifies an AS-PATH attribute containing AS 35 and AS 50 (in this order), enter:

```
ADD -BGP AsFilter 2 "_35_.*_50_"
```

Example 3 To create filter 3 that identifies an AS-PATH attribute containing AS 35 and AS 50 (in any order), enter:

```
ADD -BGP AsFilter 3 "_35_.*_50_|_50_.*_35_"
```

The horizontal bar (|) indicates a logical OR operation.

Example 4 To create filter 4 that identifies an AS-PATH attribute containing the AS Sequence <AS5, AS46, AS32>, enter:

```
ADD -BGP AsFilter 4 "<_5_ _46_ _32_>"
```

Example 5 To create filter 5 that identifies an AS-PATH attribute containing the AS Set [AS5, AS32, AS46], enter:

```
ADD -BGP AsFilter 5 "[_5_ _32_ _46_]"
```

Deny Filters Examples This section provides examples of deny filters.

Example 1 To block the import of routes containing AS 25 from all peers using filter 1, enter:

```
ADD -BGP AsPolicyAll 1 Deny In
```

Example 2 To block the advertisement of routes containing AS 35 and AS 50 to external peers using filter 2, enter:

```
ADD -BGP AsPolicyExt 2 Deny Out
```

Example 3 To block the import and advertisement of routes containing AS 35 and AS 50 to peer 10.0.0.2 using filter 2, enter:

```
ADD !10.0.0.2 -BGP AsPolicyPeer 2 Deny Both
```

Permit Filters Examples This section provides examples of permit filters.

Example 1 To permit the import of routes containing AS 25 from all peers using filter 1, enter:

```
ADD -BGP AsPolicyAll 1 Permit In
```

Example 2 To permit the advertisement of routes containing AS 35 and AS 50 to external peers using filter 2, enter:

```
ADD -BGP AsPolicyExt 2 Permit Out
```

Example 3 To permit the import and advertisement of routes containing AS 25 to peer 10.0.0.2 using filter 1, enter:

```
ADD !10.0.0.2 -BGP AsPolicyPeer 1 Permit Both
```

Configuring AS-Path Weight Policies

You can control the route selection process by assigning a specific weight to an AS, an AS path, or a subset of an AS path. The total weight for a given route is known as the *degree of preference* for the route and is calculated by summing all the individual AS-path weight expressions assigned to the route's AS-PATH attribute. If multiple routes exist for a destination, the route with the highest degree of preference is selected by the BGP route selection process. For more information, see "Path Selection" later in this chapter.

Weight policies are only applied to incoming routing updates and help control the route selection process based on AS numbers.

To configure a AS-path weight policy, follow these steps:

- 1 Define the filter to which the policy will apply using:

```
ADD -BGP AsFilter <AsfilterID> "<regular expression>"
```

For example, the following commands create filters for AS 100, 200, 300, 400, and 500:

```
ADD -BGP AsFilter 1 "_100_"
ADD -BGP AsFilter 2 "_200_"
ADD -BGP AsFilter 3 "_300_"
ADD -BGP AsFilter 4 "_400_"
ADD -BGP AsFilter 5 "_500_"
```

Blank spaces are represented in the examples as underscores (_). When two spaces are shown together, a space has been inserted between the underscores, for example __. You must enter a blank space for each underscore shown in the examples.

- 2 Define the policy (which peer the policy applies) and the weight using:

```
ADD -BGP AsPolicyAll <AsfilterID> [Weight <weight>]
ADD -BGP AsPolicyExt <AsfilterID> [Weight <weight>]
ADD -BGP AsPolicyInt <AsfilterID> [Weight <weight>]
ADD [!<IP address>] -BGP AsPolicyExt <AsfilterID> [Weight <weight>]
```

For examples of weight filters, see "Weight Filters Examples" later in this chapter.

In addition, the following syntaxes affect weight-based policies and the degree of preference in the route selection process:

```
SETDefault -BGP DefaultWeight = <number>(-2000 to 2000)
SETDefault [!<IP address>] -BGP PeerWeight = <weight>(-2000 to 2000)
```

The DefaultWeight parameter configures a default weight that is added to each route when computing the degree of preference (LOCAL-PREF attribute) for the route. You can configure this parameter to give priority in the route selection process to routes received by one BGP speaker over routes received by other BGP speakers. The default value of this parameter is 0. For information about the LOCAL-PREF attribute, see "LOCAL-PREF" later in this chapter.

The PeerWeight parameter configures a weight that is added to all routes received from the specified peer. The default value of this parameter is 0.

For more information, see “Degree of Preference Calculations” later in this chapter.

- 3 Restart all BGP sessions to recompute the route selection process by entering:

```
SETDefault -BGP CONTROL = Enable
```

Weight Filters Examples This section provides examples of weight filters.

Example 1 To assign a weight of 10 to routes with AS 100 in the AS-PATH (filter 1) received from all peers, enter:

```
ADD -BGP AsPolicyAll 1 Weight 10
```

You can assign a weight of 10 to routes with AS 100 in the AS-PATH (filter 1) received from external peers or internal peers by using the AsPolicyExt and AsPolicyInt parameters, respectively.

Example 2 To assign a weight of 10 to routes with AS 100 in the AS-PATH (filter 1) received from peer 10.0.0.2, enter:

```
ADD !10.0.0.2 -BGP AsPolicyPeer 1 Weight 10
```

Degree of Preference Calculations The degree of preference, which must always be greater than or equal to 0, is calculated before the route selection process using the following formula:

$$\text{Degree of preference} = (\text{Total ASPolicy Weight}) + \text{PeerWeight} + \text{DefaultWeight}$$

In the following examples, assume that AS 100 has a weight of 10, AS 200 has a weight of 20, AS 300 has a weight of 30, and AS 500 has a weight of 50.

Example 1 The local router receives a route with the following AS-PATH attribute: {_500_ _200_ _600_ _100_ _300_}. Assume that both the PeerWeight and DefaultWeight parameters have a value of 0. The degree of preference is equal to:

$$50 + 20 + 0 + 10 + 30 + 0 (\text{PeerWeight}) + 0 (\text{DefaultWeight}) = 110$$

If an AS in the AS-PATH attribute has not been assigned a weight using the AsPolicyXXX parameter, it is assumed to have a weight of zero (0).

Example 2 Assume that peer 10.0.0.2 has been assigned a PeerWeight of 100 and the local router's DefaultWeight value is 0. The local router receives a route from peer 10.0.0.2 with the following AS-PATH attribute: {_500_ _200_ _100_}. The degree of preference is equal to:

$$50 + 20 + 10 + 100 (\text{PeerWeight}) + 0 (\text{DefaultWeight}) = 180$$

Example 3 Assume that peer 10.0.0.2 has been assigned a PeerWeight of 100 and the local router has been configured with a DefaultWeight value of -50. The local router receives a route from peer 10.0.0.2 with the following AS-PATH attribute: {_500_ _200_ _100_}. The degree of preference is equal to:

$$50 + 20 + 10 + 100 (\text{PeerWeight}) - 50 (\text{DefaultWeight}) = 130$$

For more information, see “AS-Path-Based Policies” later in this chapter.

How the IP Router Works

This section describes the following concepts involved in IP routing activities:

- Understanding IP network topology

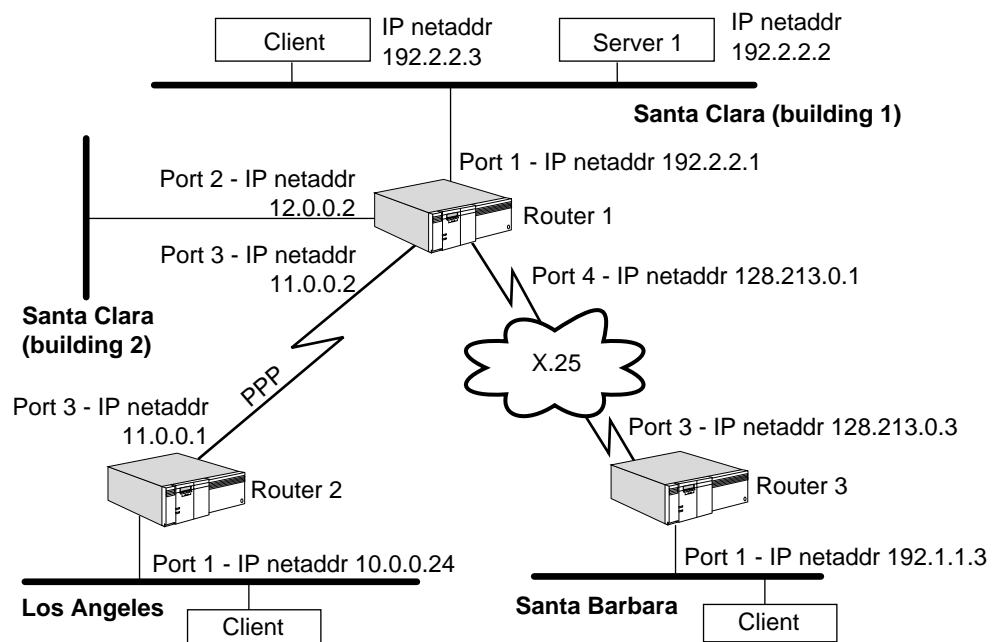
- Multipath routing
- Default routes
- Learning routes within an autonomous system
- Configuring IISIS for dual IP and OSI mode
- Learning routes between autonomous systems using BGP
- Address resolution
- Other global routing configurations

Understanding IP Network Topology

An IP network is configured on each interface where IP packets are received and sent. The interface can be either a local LAN interface or a serial line interface for a wide area network. Figure 65 shows a wide area router (Router 1) connecting two local Ethernet networks (Santa Clara buildings 1 and 2) to two wide area networks (Los Angeles and Santa Barbara). The Los Angeles network is connected by a point-to-point line, and the Santa Barbara network is connected by an X.25 link.

Although Figure 65 shows that the wide area ports that connect the Santa Clara network to the Los Angeles network are assigned IP addresses, PPP does not require that you assign an IP address to each wide area port. If you do not want to assign an IP address to a wide area port, you must set the SETDefault -IP NETaddr command to UnNumbered. For more information on this topic, see “Configuring for Local Area Networks and Point-to-Point Links” earlier in this chapter.

Figure 65 Wide Area Router Connecting Four IP Networks



CAUTION: Each IP address that you assign directly to a port must be unique, that is, you cannot assign the same IP address to different ports. If you want to give several ports the same IP address, define a port group containing the ports, and assign the IP address to the group. For information about defining port groups, see “Configuring Multiple Logical Networks” in the *Configuring Advanced Ports and Paths* chapter, and “Configuring Logical Networks over IP” earlier in this chapter.

A local network is referred to as an *attached network*. When two wide area routers are connected by one or more serial lines, their serial interfaces should be on the same network. For example, in Figure 65, port 3 of Router 1 and port 3 of Router 2 are on the same network.

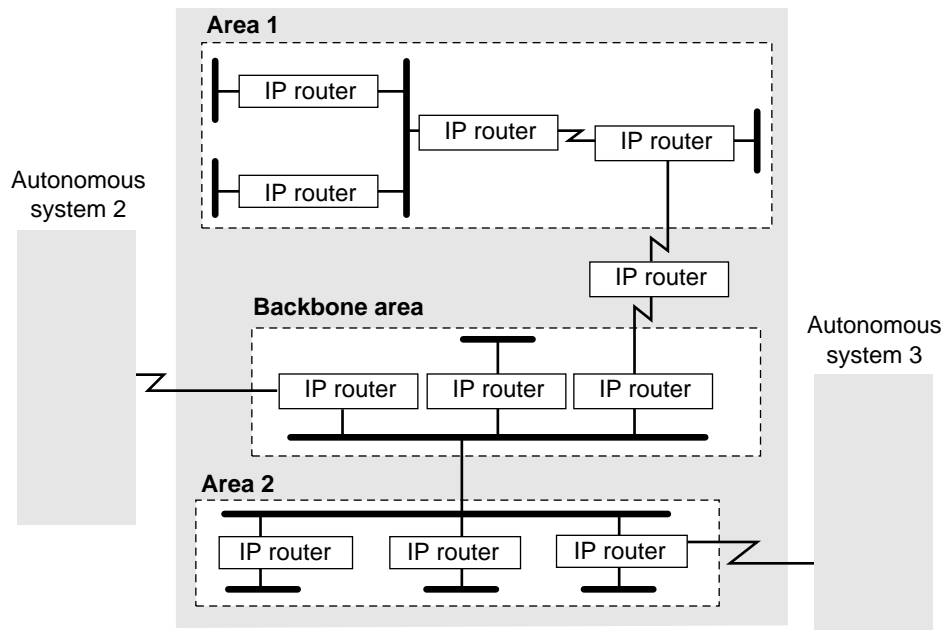
As an IP network grows, contiguous routers can be grouped into areas if using OSPF. Two areas can be interconnected through a backbone area. Areas and backbone areas can be grouped into autonomous systems. An autonomous system consists of routers and networks administered by a single authority. An autonomous system typically runs a single intra-autonomous system routing protocol, such as OSPF.



RIP does not support areas.

Routing can take place between autonomous systems using an interautonomous system protocol, such as the Border Gateway Protocol (BGP). Figure 66 shows two areas within an autonomous system being connected by a backbone area. It also shows an autonomous system connected to two other autonomous systems.

Figure 66 Typical IP Network Running OSPF
Autonomous system 1



A router must check its routing table to determine where to route a packet. If the destination is on an attached network, the router can send it directly to the network. But if the destination is farther away in the internetwork, the router must route the packet to another router (called a *gateway*) that is closer to the destination. The route to a wide area network can be statically configured or dynamically learned through RIP, OSPF, IISIS, and BGP. When two routers are located on the same network (that is, each of them has at least one interface to the network), they are considered *neighbors* or *neighboring gateways*.

Multipath Routing

The router supports multipath routing, which means that up to four routes for each destination address can be stored in the routing table. Advantages of multipath routing are as follows:

- The router can still route a packet using an alternative route if the primary one fails; it is more responsive to network topology changes than if only one route to a destination exists.
- The router can distribute the load among the available equal-cost best paths.

Routes learned by different routing protocols are assigned a different precedence in the routing table. Some routing protocols such as OSPF and IISIS have multiple route classes. The classes are also assigned a different precedence.

When multiple routes for a destination exist, the router uses the route with the highest precedence. The types of routes used (listed by decreasing precedence) are listed in Table 15.

Table 15 Route Precedence

Precedence Level	All Protocols without BGP	All Protocols with BGP
1	Static (added without Override)	Static (added without Override)
2	OSPF Intra Area	OSPF Intra Area
3	OSPF Inter Area	OSPF Inter Area
4	IISIS Intra Area	IISIS Intra Area
5	IISIS Inter Area	IISIS Inter Area
6	RIP	RIP
7	OSPF Type 1 External	OSPF Type 1 External
8	OSPF Type 2 External	OSPF Type 2 External
9	IISIS External	IISIS External
10	ICMP redirect (not applicable for a router; only for host mode)	ICMP redirect (not applicable for a router; only for host mode)
11	Static (with override)	Static (with override)
12a		BGP
12b		Route configured with DefaultNets
13		Non-BGP default route

When the IP Protocol routes a packet and BGP is enabled, the software looks up the route as follows:

- 1 Looks for a route learned by any protocol except BGP in the All Protocols Routing Table.
 - The software searches in the order specified in the “ All Protocols without BGP” column in Table 15.
 - BGP has its own routing table separate from all the other protocols.
 - The software does not consider the default route yet.
- 2 If a route to the destination is not found, the software looks for the route in the BGP Routing Table (precedence level 13a in Table 15).

- 3 If a route to the destination is not found in the BGP Routing Table, the software looks for any configured default networks configured with the `ADD -BGP DefaultNet <IPAddress>` syntax.
- 4 If no default networks are configured, the software again searches the All Protocols Routing Table, looking for default routes.

If BGP is disabled, the software follows steps 1 and 4.

The routing table displays routes with a high precedence first.

OSPF Type 1 and 2 external metrics allow you to define how you want Autonomous System Boundary Routers (ASBRs) to report metrics.

A Type 1 external metric is the sum of the metric learned within the autonomous system by OSPF plus the metric learned outside of an autonomous system by BGP. A Type 2 external metric is the metric learned outside of an autonomous system by BGP only.

In an OSPF environment, you can set your ASBR to report Type 1 or 2 external metrics using the `InteriorPolicy`, `ExteriorPolicy`, `Static Policy`, and `DefaultMetric` parameters. For complete information on these parameters, see *Reference for Enterprise OS Software*.

If the route with the highest precedence fails, the route with the next highest precedence will be used. A route in the routing table is deleted in these situations:

- OSPF and IISIS both compute routes based on link state information from all routers. The entire routing table is recomputed each time the topology changes.
- A dynamic route learned through RIP is deleted when a router times out and goes through the HOLD-DOWN and GARBAGE COLLECTION states. A router times out when it fails to hear from a neighbor for a period that is six times the value of the `UpdateTime` parameter. For example, if the value of the `UpdateTime` parameter is 45 seconds, the router will time out if it does not hear from its neighbor for 270 seconds.
- A `DELeTe ROUte` command removes a static route.
- A lowest precedence route is deleted when four routes of higher precedence are available. This situation occurs when a fifth route is learned and has a higher precedence than the lowest precedence route.

Dynamic routes learned by RIP can be removed by using the `FLush -IP AllRoutes` command. However, this command does not flush routes learned by OSPF, IISIS, or BGP from the routing table.

Route Selection and Load Splitting

If two or more routes with the same route source precedence are available to reach a destination, the router always selects the route with the lowest metric (measured in hops for RIP and in administrative cost for OSPF, or IISIS). If there is more than one route learned by the same routing protocol with the same equal-cost, low metric, you can split the load between these routes on a round-robin basis. The `-IP CONTrol` parameter (`SplitLoad` | `NoSplitLoad`) determines whether load splitting is performed.

Because load splitting balances the load among different routes, 3Com recommends it if two or more routes are available to reach a destination and the routes have similar metrics. However, if the routes connecting various networks have different metrics (that is, there is only one route with the fewest hops or lowest cost to a destination), load splitting is not necessary.

Route Selection Examples

Table 16 is an example of a routing table and shows how a route is chosen.

Routes are selected on the basis of precedence, lowest metric, or in cases where multiple routes have the same precedence and metric, through load splitting or the first route discovered. The examples in Table 16 demonstrate these criteria and can be applied to all types of routes.

Table 16 Routing Table Containing Multiple Paths

Network	Gateway	Metric*	Route Source
10.0.0.0	129.213.1.1	100	OSPF—Intra
	129.213.1.2	1	RIP
20.0.0.0	129.213.16.1	1	RIP
	129.214.1.1	2	RIP
30.0.0.0	129.213.16.1	100	OSPF—Intra
	129.213.16.2	100	OSPF—Intra
	129.213.16.3	100	OSPF—Intra

* RIP uses hop count as its metric. The OSPF metric is computed from total administrative cost between router and destination.

Example 1 For network 10.0.0.0, there are two routes available, but these routes are not comparable. The first route is learned by OSPF, and the second route is learned by RIP. Because routes learned by OSPF take precedence over routes learned by RIP, gateway 129.213.1.1 is selected.

Example 2 For network 20.0.0.0, there are two routes available through RIP. Because gateway 129.213.16.1 requires one hop and gateway 129.214.1.1 requires two, the router always selects gateway 129.213.16.1 because it requires the fewest hops or the lowest metric to reach its destination.

Example 3 The routing table entry for network 30.0.0.0 has three available routes to reach it. All are dynamic routes learned through OSPF, and all require an administrative cost of 100. The router chooses the route as described here:

With load splitting The route is chosen on a round-robin basis. Gateway 129.213.16.1 is used first, then 129.213.16.2, then 129.213.16.3. If one of these routes becomes invalid, it is no longer considered in the selection procedure.

Without load splitting The route recorded earliest is always used. In this case, the gateway 129.213.16.1 is used.

Default Routes

When a router needs to route a packet destined for an address for which there are no entries in the routing table, it uses the default route if one exists. The network 0.0.0.0 represents the default route. The router supports up to four default routes;

when more than one default route is available, the same selection rules apply. If load splitting is enabled, the load is distributed among equal-cost best paths. For additional information, see “Multipath Routing” earlier in this chapter.

An advantage of a router using a default route is that network overhead in an autonomous system can be reduced. The reduction in overhead occurs because the router does not need to advertise all external routes.

The following example will help you understand default routes.

Example Router A receives a RIP update packet from router B, which has an entry indicating that network 0.0.0.0 is reachable with metric 3. Router A considers router B its default gateway. That is, if router A needs to route a packet whose destination is not found in its routing table, it sends the packet to router B.

The interior routing protocols for IP (RIP, OSPF, and IISIS) can be configured to advertise a default route by assigning a non-zero value to the `DefaultMetric` parameter of the routing protocol's service. You do not need to configure the `DefaultMetric` parameter on every router throughout the domain. The default route learned on one interface is propagated to neighbors on the other interfaces (unless inhibited by the `NetworkPolicy` parameter).

Each interior routing protocol propagates the advertisement of the default route as the normal operation. For the RIP Protocol, it is possible to suppress propagation of the default route by using the `AdvertisePolicy` parameter. Since OSPF and IISIS are both link state routing protocols, they cannot suppress any routing information within the bounds of their routing system. However, they can control information that they import from other routing systems.

If more than one interior routing protocol is in operation on a network, the routes from one system can be introduced into the other system by using the `InteriorPolicy` parameter of the protocol that is importing the routes.

Suppose that RIP and IISIS are both in operation, and that RIP needs to import routes from the IISIS system. Use the `InteriorPolicy` parameter in the `RIPIP Service` to achieve this routing. For more information on this parameter, see the `RIPIP Service Parameters` chapter in *Reference for Enterprise OS Software*.

When a router is operating both an interior protocol (RIP, OSPF, or IISIS) and an exterior protocol (BGP), then the `ExteriorPolicy` parameter in the service of the interior routing protocol is used to control the import of routes from the exterior routing protocol into the interior routing protocol. The `InteriorPolicy` parameter in the service of the exterior routing protocol is used to control the import of routes from the interior routing protocol into the exterior routing protocol.

You can configure the default route in one of two ways:

- On the exit router of a domain, configure a static override default route with a metric of 1 that points to the first hop outside the domain. Then use the `StaticPolicy` parameter of the selected routing protocol to import this route into the routing protocol and advertise it into the domain.
- At the top level, set the `DefaultMetric` parameter in the selected routing protocol to instruct the router to originate a default route.



The second method may also be useful at a router that interconnects a RIP domain with an OSPF domain, instead of importing all routing information from each domain into the other.

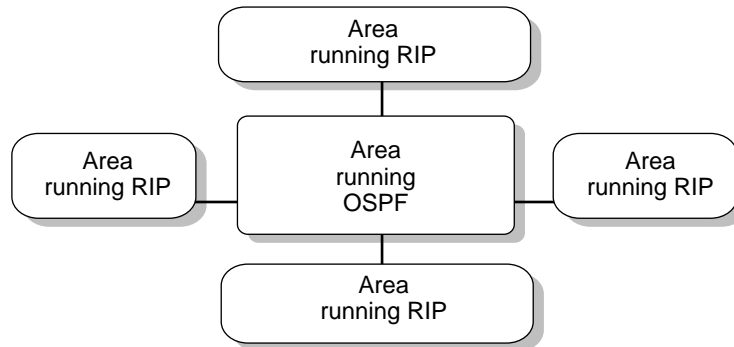
If the bridge/router is instructed to originate the default route (by setting the DefaultMetric parameter to a nonzero value), it does not accept another router's advertisement of the default route. Consider two routers in parallel, both originating a default route and each accepting the other's advertisement of the default route. Any packet received by one router is forwarded to the other router, and back again, until the time-to-live timer is exhausted and the packet is dropped.

Learning Routes within an Autonomous System

The router fully supports RIP according to RFC 1058. It also supports OSPF Version 2 according to RFC 1583.

If you are planning to use both RIP and OSPF when expanding your IP network, 3Com recommends that autonomous systems using OSPF make up the core of the network and that autonomous systems using RIP surround those using OSPF. Figure 67 is an example of this topology.

Figure 67 Recommended Autonomous System Topology



If a router runs both OSPF and RIP Protocols, the routes learned by one of these protocols are not reported by the other according to the default settings of the InteriorPolicy parameter in both the RIP and OSPF Services. If you want cross-reporting between these protocols, set the InteriorPolicy parameter in the RIP, ISIS, and OSPF Services accordingly.

If you are using OSPF in a topology with end stations, you need to configure a default gateway on the end stations. Many end stations learn RIP routes dynamically, but they usually do not learn OSPF routes dynamically.

Learning Routes with RIP

Normally, every 30 seconds or every time it learns a route change for a network, the router uses broadcast packets to report to its neighboring gateways the following types of information:

- The networks it can reach
- The metric associated with each network it can reach

By default, the information in update packets pertains only to learned routes. Static route information is not reported.

You can configure some router parameters (see “ User Configurations” later in this chapter) to determine how the router sends out the updates and what is included in them. For example, you can configure the parameters for the following purposes:

- To change the frequency of the broadcast traffic (UpdateTime parameter)
- To prevent the router from sending or receiving update and request packets (CONTRol parameter)
- To control the set of neighboring routers from which the router receives updates and to which it sends them (AdvToNeighbor and RcvFromNeighbor parameters)
- To prevent the router from sending out a trigger update response upon a route change for a network (CONTRol parameter)
- To enable the router to report static routes (StaticPolicy parameter)
- To enable the router to report routes learned with other interior routing protocols, such as IISIS and OSPF (InteriorPolicy parameter)
- To cause some routes not to be reported or to be reported with the infinity metric, that is, using poison reverse (CONTRol parameter)

Network Reachability

The following types of networks are considered *reachable* when a router broadcasts its RIP update packets:

- All directly connected networks, unless the network is shared by its neighbor and itself
- All static routes (as controlled by the StaticPolicy parameter)
- All dynamic routes learned through RIP and either OSPF or IISIS in the routing table (as controlled by the InteriorPolicy parameter)
- All dynamic routes learned through BGP (as controlled by the ExteriorPolicy parameter)

Solving the Slow Convergence Problem with Split Horizon

Ideally, all routers learn of new routes and discard obsolete routes immediately. That is, the contents of their respective routing tables converge rapidly so that all routing tables always contain correct information. An undesirable side effect of RIP is the possibility that the time is prolonged during which the unreachable network is considered reachable. One solution to this problem of slow convergence is called *split horizon*.

In a WAN environment, the 3Com implementation of next-hop split horizon (-RIPIP CONTRol = NonMesh) eliminates the need for a fully meshed network when using RIP. In next-hop split horizon, the router learning of a network records the IP address of the neighbor from which the network was learned instead of recording the port through which the network was learned. When the router advertises its

own reachable networks, it advertises to all neighbors except the one from which it learned of the network being advertised.

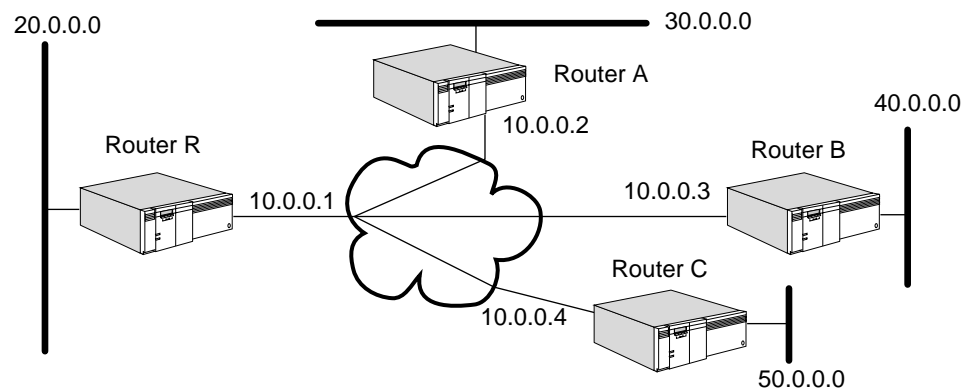
Nonmeshed WAN Networks Figure 68 shows a nonmeshed Frame Relay, X.25, or ATM network using RIP on which router R is the root router and routers A, B, and C are remote routers that are configured as neighbors on router R. Router R sends RIP updates individually to its neighbors, remote routers A, B, and C. When sending RIP packets, router R advertises to neighbors all networks it knows about (in this example, networks 20.0.0.0, 30.0.0.0, 40.0.0.0, and 50.0.0.0) if next-hop split horizon is not used. Network 10.0.0.0, being common to all routers in the diagram, is automatically excluded from RIP updates between these routers.

By applying next-hop split horizon, router R does not advertise network 30.0.0.0 to router A, because it learned of 30.0.0.0 from router A. Router R also does not advertise network 40.0.0.0 to router B, nor does it advertise 50.0.0.0 to router C, because it learned of those networks from those routers.

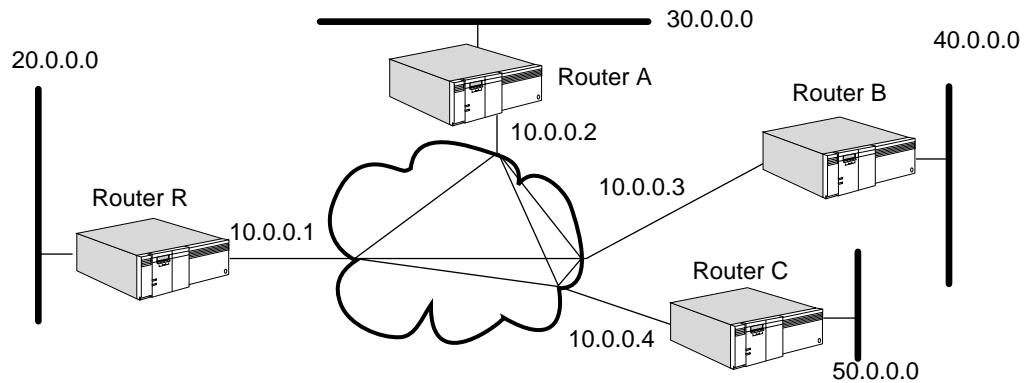


You need to enable the next-hop split horizon feature by setting -RIPIP CONTROL to NonMesh.

Figure 68 Route Advertisement over Nonmeshed Frame Relay or X.25 Network



Meshed WAN Networks In Figure 69, the WAN network is meshed because all routers are directly connected to one another. Even if a WAN network is meshed, you must configure routers A, B, and C as neighbors on router R, the root router, for RIP to unicast updates over the WAN. You also need to set -RIPIP CONTROL to FullMesh so that next-hop split horizon is disabled. This example applies to Frame Relay, ATM, and X.25 networks. With Frame Relay networks, RIP neighbors can be dynamically learned.

Figure 69 Route Advertisement over Meshed Frame Relay Network

LAN Networks On a LAN network, it is not necessary to configure neighbors. If you do not configure neighbors, RIP broadcasts the updates over the LAN. If you configure neighbors, RIP unicasts the updates.

Solving the Slow Convergence Problem with Poison Reverse

Poison reverse or no poison reverse is configurable using the Poison or NoPoison value for the -RIPIP CONTROL parameter.

If poison reverse is enabled, the router advertises all routes to all neighbors, but when advertising a route to a neighbor that has advertised the same route, the router sets the metric to infinity (0xFFFF) to prevent the recipient from adding the route to its routing table. Poison reverse speeds convergence but adds to network overhead.

If poison reverse is disabled, the router omits routes learned from one neighbor from RIP updates sent to that neighbor. No poison reverse has the advantage of minimizing network overhead in large network configurations at the expense of slower convergence.

User Configurations

Table 17 shows how you can change the way the router broadcasts or processes RIP update packets. This table includes only the parameters that were not discussed in previous sections. For complete information on the parameters listed in this table, see the RIP Service Parameters chapter in *Reference for Enterprise OS Software*.

Table 17 Configuring the IP Router for RIP Updates Using RIPIP Parameters

Parameter	Result
UpdateTime*	Changes the frequency of the update packets.
CONTROL parameter option:	
TRigger NoTRigger	Determines whether a route change for a network triggers an update packet from the router.
AdvToNeighbor	Determines to which gateways on the directly connected networks the router sends the update packets.

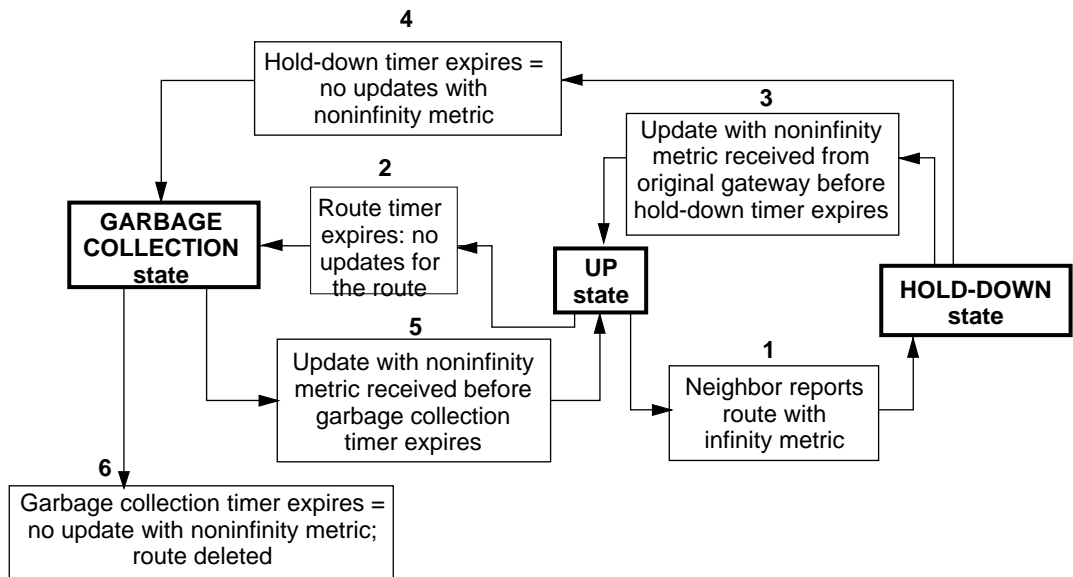
Table 17 Configuring the IP Router for RIP Updates Using RIPv2 Parameters

Parameter	Result
RcvFromNeighbor	Determines to which gateways on the directly connected networks the router should listen for routing information.

* The parameter applies to the entire router. All routers exchanging RIP information should have the same value for this parameter. Otherwise, routing loops or loss of connectivity in the network may occur.

Different States of RIP-Learned Routes

To avoid routing loops, new information about a route is ignored for a designated period before it is used. Figure 70 summarizes how a route learned through RIP changes states. Explanations of the different states follow the figure.

Figure 70 Different States of a RIP Route

- **GARBAGE COLLECTION state**

When the timer for a route that has been in the HOLD-DOWN state expires, that route changes to GARBAGE COLLECTION state. This happens when no update packets are received to indicate that the route is still reachable. In this state, if a neighboring gateway reports the route with a noninfinity metric within 120 seconds, the route can go back to the UP state. If no updates are received within 120 seconds (garbage-collection timer), the route is deleted from the routing table. It is possible to go into GARBAGE COLLECTION state if no updates are received within 180 seconds.

- **UP state**

A route is considered UP if it is reachable with a noninfinity metric (15 or fewer hops). Whether it is reachable is determined by the last update received from the neighboring gateways. It remains UP for 180 seconds (the route timer). The timer is reset each time a new update for the route is received.

- **HOLD-DOWN state**

A route in UP state changes to HOLD-DOWN state if an update received from the original gateway indicates that the route is associated with an infinity

metric (16 hops). In this state, all update information received from other gateways for that route is ignored.

However, if an update is received from the original gateway within 60 seconds (the hold-down timer), and it associates a noninfinity metric with the route, the route goes back to UP state.

If the hold-down timer expires, the route goes from HOLD-DOWN state to GARBAGE COLLECTION state for 120 seconds.

When you display the routing table with the `SHoW -IP AllRoutes` command, the state of each route is displayed under the STATUS heading.

Learning Routes with OSPF

Normally, every 30 minutes or every time the router learns a route change for a network, it uses multicast packets to report to its neighbors the following types of information:

- The networks and the directly connected routers
- The metric associated with each directly connected router and network

In an unchanging topology, OSPF only sends updates every 30 minutes while RIP sends updates every 30 seconds. OSPF provides a significant savings in network overhead when compared to RIP.

Different Functions of OSPF Routers In an autonomous system running OSPF, routers can be assigned several different functions. An OSPF router can be assigned to route within an area (intra-area), between areas (interarea), or between autonomous systems (interautonomous system).

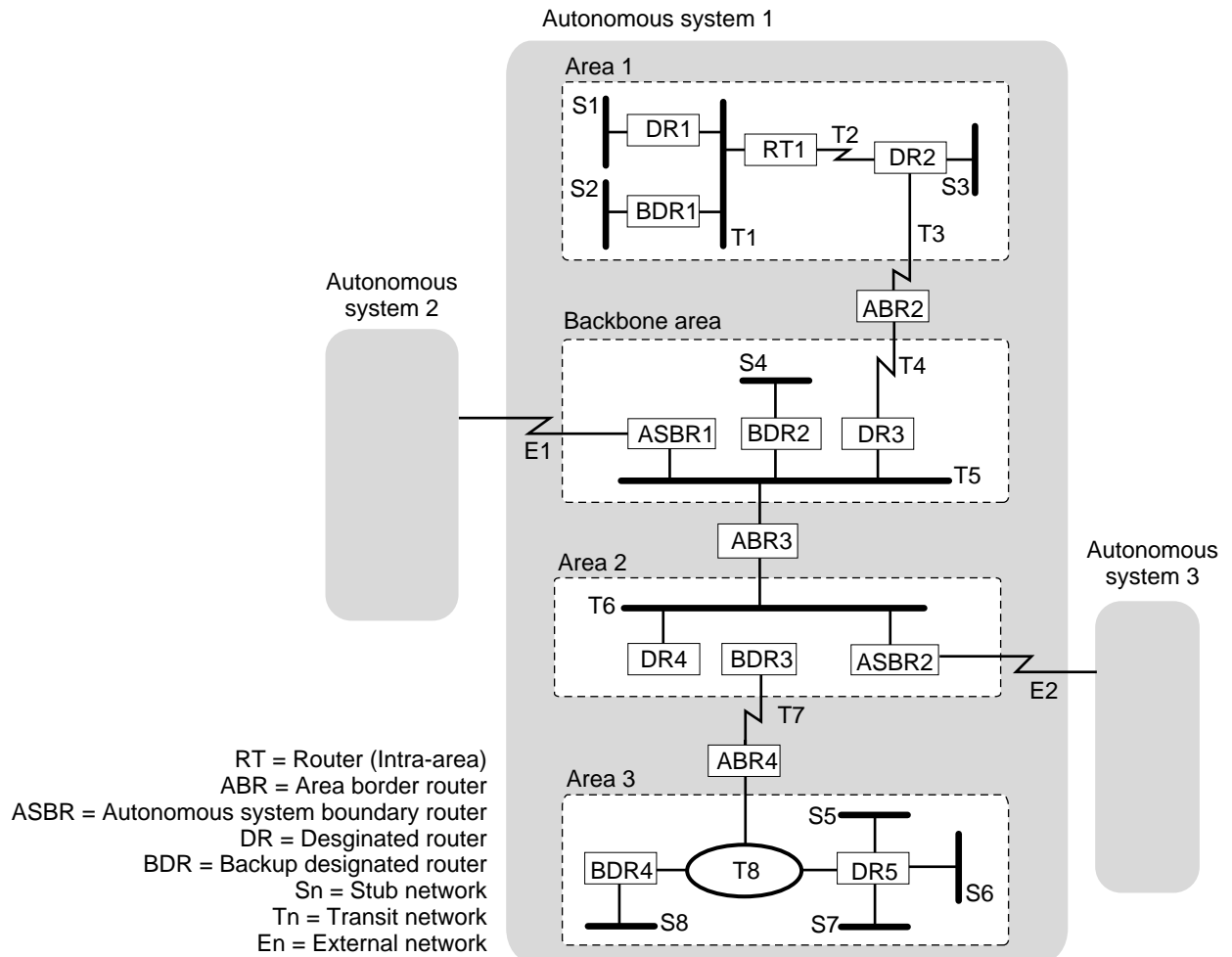
By default, the router performs intra-area routing. A router that routes between areas is an Area Border Router (ABR). Routing between autonomous systems is performed by a router that acts as an Autonomous System Boundary Router (ASBR).

In addition to its routing function, a router can function as the designated router (DR) or backup designated router (BDR) on a multiaccess network. (A multiaccess network is any network other than a point-to-point link, such as SMDS, X.25, Frame Relay, or a LAN.)

Figure 71 is an example of an autonomous system running OSPF, with routers configured as described in the preceding paragraphs. Detailed descriptions of

ABRs, ASBRs, DRs, and BDRs follow the figure. A stub network is a network that only has one OSPF router; a multiaccess network has more than one OSPF router.

Figure 71 Autonomous System with Multifunctional OSPF Routers



Area Border Router An area border router (ABR) is a router that has interfaces in more than one area. For example, in Figure 71, ABR2 interfaces network T3, which is part of Area 1. It also interfaces network T4, which is part of the backbone area. (A router automatically acts as an ABR when different area numbers are assigned to different ports.)

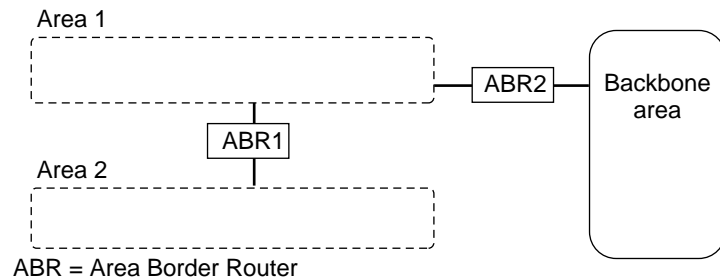
Each ABR maintains a distinct database for each area to which it belongs. In the example shown in Figure 71, ABR2 maintains databases for Area 1 and the backbone area.

All ABRs should have at least one interface connected to the backbone area. However, if there are no interfaces of an ABR connected to the backbone area, you can configure a virtual link to provide complete connectivity.

A virtual link is established between two ABRs. One of the ABRs must be directly connected to the backbone area, which provides a link for the other ABR. Also, both ABRs must be part of at least one common nonbackbone area for the virtual link to be established.

For example, in Figure 72, ABR1 is an ABR for Areas 1 and 2. However, it is isolated from the backbone area. Because ABR1 and ABR2 are both connected to Area 1 and ABR2 is connected to the backbone area, a virtual link can be established between ABR1 and ABR2. The VirtualLink parameter allows you to establish a virtual link between two ABRs. For more information on this parameter, see the OSPF Service Parameters chapter in *Reference for Enterprise OS Software*.

Figure 72 Backbone Area with One Isolated ABR



Autonomous System Boundary Router An autonomous system boundary router (ASBR) is a router that interfaces one or more routers in other autonomous systems. For example, in Figure 71, ASBR1 interfaces networks in Autonomous Systems 1 and 2. (A router automatically acts as an ASBR when different routing protocols [RIP, OSPF, IISIS, or BGP] are enabled on different ports.)

An ASBR can also function as an ABR if it is connected to more than one area in addition to being connected to another autonomous system.

Typically, an ASBR runs an interautonomous system routing protocol, such as BGP, on the interface that connects the other autonomous systems. On interfaces within the autonomous system that it is part of, the ASBR runs an intra-autonomous routing protocol, such as RIP, OSPF, or both.

In addition to its routing function, a router that is elected as the designated router (DR) on the multiaccess network performs the function of flooding for the network. A router that is elected as the backup designated router (BDR) on the multiaccess network should be adjacent to the same routers that the DR is adjacent to. The BDR has a subset of the DR's responsibilities. The BDR takes over the DR's role in the event of its failure. For details on the functions performed by the DR and the BDR, see "Learning Routes and Network Reachability" next.

The value of the ROUTerPriority parameter determines which routers on a multiaccess network function as the DR and the BDR. For more information on the ROUTerPriority parameter, see the OSPF Service Parameters chapter in *Reference for Enterprise OS Software*.

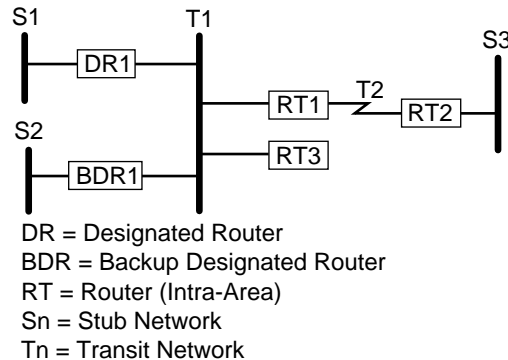
Learning Routes and Network Reachability In an autonomous system running OSPF, intra-area routes, interarea routes, and interautonomous system routes are learned as described here.

Each router periodically exchanges a hello packet with its neighbor. The hello packet includes a list of all routers from which the originating router has recently received a hello packet. The exchange of hello packets establishes a bidirectional relationship between neighbors. The HelloTime parameter allows you to set the

frequency at which hello messages are sent. If you modify the setting of the HelloTime parameter, you must check the setting of the RouterDeadTime parameter. If the setting of the HelloTime parameter is larger than the setting of the RouterDeadTime parameter, the routers will not become fully adjacent. For more information on the HelloTime and RouterDeadTime parameters, see the OSPF Service Parameters chapter in *Reference for Enterprise OS Software*.

After bidirectional relationships are established between neighboring routers, each pair of neighboring routers must decide if they should form an adjacency (a formalized bidirectional relationship). Neighboring routers connected by a point-to-point network always form adjacencies. (A point-to-point network is a network where two routers are connected through a single network connection.) However, on a multiaccess network, adjacencies are formed only between DRs and BDRs and each of their neighbors. For example, in Figure 73, DR1 is the DR and BDR1 is the BDR for Network T1. Adjacencies are formed between DR1 and RT1, between BDR1 and RT1, between DR1 and RT3, between BDR1 and RT3, between RT1 and RT2, and between DR1 and BDR1. However, RT1 and RT3 have not fully established an adjacency with each other. They are in a state known as a two-way state.

Figure 73 Forming Adjacencies on a Multiaccess Network



If adjacencies were formed between each router and its neighbor on a multiaccess network, it can be shown mathematically that the amount of traffic on the network would be significantly heavier. By minimizing these adjacencies, the DRs and BDRs reduce this traffic to manageable proportions.

Use the `SHoW -oSPF NeighborStatus` command to display an OSPF neighbor status table, which shows the status of direct connect neighbor adjacencies for your router. For more information on the `SHoW NeighborStatus` command and an explanation of OSPF neighbor status table entries, see the OSPF Service Parameters chapter in *Reference for Enterprise OS Software*.

After an adjacency is formed between a pair of routers, each router on a point-to-point network and the DRs and BDRs on a multiaccess network send out a link state advertisement to its neighbor every 30 minutes or whenever a change in topology occurs. External link state advertisements are flooded throughout the router's autonomous system. Other link state advertisements are flooded within a

single area. For details on link state advertisements, see “Link State Advertisements” later in this chapter.

Each router in an area maintains an identical database of the area’s topology. The database contains both the topology of the router’s area and routes to networks outside of the router’s area. This database is used to build a shortest path tree. The router doing the computation uses itself as the root of the tree and builds each node of the tree based on the metric advertised in the link state advertisement.

The router always selects the path with the lowest metric. For details on metrics, see “Metrics” later in this chapter.

If there is more than one equal cost path, the router can use multipath routing and load splitting. For more information on these features, see “Multipath Routing” and “Route Selection and Load Splitting” earlier in this chapter.

The router stores information on all reachable networks in its routing table.

The following types of networks are considered reachable:

- All directly connected networks, unless the network is shared by its neighbor and itself
- All static routes (if configured)
- All dynamic routes learned through RIP, OSPF, and IISIS in the routing table
- All dynamic routes learned through BGP (if configured)

For more information, see descriptions later in this chapter.

Reducing Network Costs Using Demand Interface Circuits In a remote office internetworking environment, many remote offices are connected to a central site through *demand circuits*, such as ISDN circuits or analog lines with modems, X.25 SVC or Frame Relay SVC neighbors, or dial-up lines. The cost of these demand circuits depends on the connection time or line usage.

OSPF periodically sends hello packets to refresh routing information, requiring the circuit to be constantly open, which results in unwanted usage charges. To reduce the cost of running OSPF over demand circuits and increase bandwidth, OSPF has been modified to operate more efficiently over demand circuits. When no network topology changes occur, OSPF sends no routing information traffic at all, allowing the data link connection to be closed when not required for application data traffic. As soon as data is sent, a data link connection is attempted. If the connection is successful, the data is sent and the circuit stays open. After a period of inactivity, the circuit is closed again to conserve cost and resources.

Using the `-OSPF DemandInterface` parameter, you can configure an interface to be a demand interface. The neighboring router must agree that the point-to-point link is a demand circuit by setting the DC bit defined in the OSPF Options field of router LSAs, OSPF hello packets, and database description packets as follows:

- In a router’s self-originating LSAs, the DC bit is set if and only if the router can properly process LSAs having the DoNotAge bit set.

If the DoNotAge bit is set, only truly changed LSAs are flooded over demand circuits. If a newly received LSA is only a periodic refresh, it is not flooded on attached demand circuits.

LSAs are not aged while they are held in the link state database, meaning they do not have to be refreshed, further reducing the routing traffic and the amount of time the circuit must remain up.



CAUTION: *Do not configure any interface on any router in a single OSPF area as a demand circuit (DC) interface unless all routers in that area have been upgraded to at least software version 8.3. Non-DC-aware routers become confused by LSAs using the DoNotAge bit in the link state age field. The LSA appears to expire and those routers are constantly flushing the LSA from their link state database and rerunning the Dijkstra algorithm, as well as informing all the routers they have adjacencies with of the routing changes. This affects every router in an area that cannot understand DC-style LSAs.*

- For hello and database description packets, the DC bit is set in outgoing packets if and only if the router wants to treat the attached network as a demand circuit and tries to negotiate with the neighboring router for the suppression of hellos on point-to-point demand circuits.

Over point-to-point demand circuits, both end points must agree to suppress sending hello packets by setting the DC bit in OSPF hellos and database description packets. Receiving a packet with this setting indicates agreement, and OSPF hello packets are sent only until initial link state database synchronization is achieved with the neighbor. After the state of the neighbor connection reaches “full,” hellos are suppressed and the data link connection to the neighbor is assumed to be available.

For OSPF broadcast and nonbroadcast multiaccess (NBMA) networks that have been configured as demand circuits, the exchange of hello packets remains periodic for the proper operation of the DR election algorithm.

Link State Advertisements A link state advertisement identifies the state of a router’s interfaces and adjacencies. The types of link state advertisements sent out by a router depends on the function that the router has been configured to perform. Multiple link state advertisements can be contained in a link state update packet. There are four types of link state advertisements:

- Router link state advertisements

Each OSPF router sends out a router link state advertisement. This advertisement describes its links to stub and multiaccess networks as well as links to other routers for a given area. (A stub network is a network that only has one OSPF router; a multiaccess network has more than one OSPF router.) The advertisement is flooded throughout the area the originating router belongs to.

- Network link state advertisements

DRs on multiaccess networks send out network link state advertisements. These advertisements describe the routers on the network that are fully adjacent with the DRs. These advertisements are flooded throughout a single area.

- Summary link state advertisements

ABRs send out summary link state advertisements. These advertisements summarize all the interarea routes for all the areas to which the router is attached. (Each available interarea route is summarized in a separate summary link state advertisement.) These advertisements are flooded throughout the area that the ABR interfaces.

- External link state advertisements

ASBRs send out external link state advertisements. These advertisements contain information on destinations outside of the autonomous system the router resides in, including static and dynamic routes learned by RIP, BGP, and IISIS. (Each destination outside the autonomous system is described in a separate external link state advertisement.) These external routes are described using Type 1 or 2 external metrics. (For more information on Type 1 and 2 external metrics, see “Metrics.”) These advertisements are flooded throughout the autonomous system the router resides in.

To view the short version of the link state database, enter:

```
SHow -OSPF LinkStateData
```

Metrics OSPF uses an administrative cost as its metric. The SETDefault -OSPF Cost command allows you to set the cost for a specific path.

The default value of the Cost parameter is 10^8 /bandwidth of the medium that interfaces a port. For Ethernet, the default value is 10; for T1 lines, the default value is 65; for FDDI, the default value is 1.

For example, to set the cost on port 3, enter:

```
SETDefault !3 -OSPF Cost = 58
```

In this example, the T1 serial line that interfaces port 3 has been assigned the cost of 58. For more information on the Cost parameter, see the OSPF Service Parameters chapter in *Reference for Enterprise OS Software*.

The router running OSPF selects the route with the lowest total administrative cost to reach its destination. For example, imagine that OSPF learns about two intra-area routes to reach a particular destination. Route 1 has the administrative cost of 200, while Route 2 has the administrative cost of 300. The OSPF router will select Route 1 because it has the lowest administrative cost.

User Configurations Table 18 summarizes the OSPF parameters that allow you to customize the configuration of your OSPF router. For complete information on these parameters, see the OSPF Service Parameters chapter in *Reference for Enterprise OS Software*.

Table 18 OSPF Configuration Parameters

Parameter	Operation
AreaID	Determines the area to which a specified port on a router belongs.
Cost	Determines metrics (cost and type of service) associated with a specified port.
(continued)	
DEBUG	Determines the level of OSPF tracing that will be performed at the local console port.
DefaultMetric	Determines the metric for the default route.
Delay	Determines the delay in seconds for the specified port. The delay time is added to all link state advertisements before it is sent on an interface.

Table 18 OSPF Configuration Parameters (continued)

Parameter	Operation
DirectPolicy	Determines whether a locally attached network should be advertised into the OSPF domain
ExteriorPolicy	Determines which networks learned through BGP are reported in external link state advertisements and what metric and metric type to use.
HelloTime	Changes the frequency at which hello packets are exchanged between neighbors on a network.
InteriorPolicy	Determines which networks learned through RIP and IISIS are reported in external link state advertisements and what metric and metric type to use.
Neighbor	Determines to which router on the directly connected network a router sends packets. Also determines how packet is addressed.
PassWord	Determines the password for a specified port to authenticate packets.
ReceivePolicy	Determines which networks from external link state advertisements are stored in routing tables and what metric to use.
RouterDeadTime	Changes the frequency for determining when a router is down.
ROUTerPriority	Determines the priority for a router on a specified port. The router with the highest priority becomes the DR for a multiaccess network.
StaticPolicy	Determines which static routes are advertised for a particular network and what metric is used.
StubDefaultMetric	Specifies whether or not the router should generate the default route and metric into the stub areas.
Virtuallink	Determines whether the specified port of a router acts as a virtual link between an area and a backbone area.

Configuring Integrated IS-IS for Dual IP and OSI Mode

If you are configuring dual IP and OSI mode, you must enable the IP forwarding process, enable Connectionless Network Protocol (CLNP), and have at least one IP network number or subnet mask configured before configuring IISIS.

IISIS is a protocol that provides integrated OSI-type routing for IP and OSI environments; it is the IP extension added to the original OSI IS-IS Protocol. IISIS routing simplifies network topology, reduces network management complexity, and reduces routing traffic overhead. In IP environments, IISIS is an alternative to other IP routing protocols, such as RIP and OSPF.

The original IS-IS routing protocol was developed by ISO to provide network layer connectivity in OSI environments. IS-IS is designed to work with CLNP and ES-IS. IS-IS is an international standard.

You can use IISIS in OSI mode for routing in pure OSI environments. You can use IISIS in dual IP and OSI mode for routing in environments where both types of networks are being used. In dual mode, for example, one router can serve IP and

OSI subnets simultaneously and IISIS routes traffic between the two subnets. You can also use IISIS for IP environments only.

Autonomous System Routing Using BGP

The Border Gateway Protocol (BGP) is an interautonomous system routing protocol that is used to exchange routing information between different autonomous systems (ASs). A router can use BGP to determine the reachability of networks outside of its AS.

The sections that follow describe the following items:

- BGP overview
- BGP external and internal peers
- Peer-to-peer communication
- Path attributes (AS-PATH, ORIGIN, NEXT-HOP, MULTI-EXIT-DISC, LOCAL-PREF, ATOMIC-AGGREGATE, and AGGREGATION)
- Path selection
- Policies (interior, exterior, network number, AS-path)
- Route aggregation

BGP Overview

The BGP provides the following advantages:

- Consumes less bandwidth

The BGP uses incremental updates to reduce the amount of routing information. When a BGP session is first established, the peers exchange the entire contents of their routing tables. After this initial data exchange, BGP peers only exchange changes to their routing tables, effectively reducing the size of their routing tables and consuming less bandwidth.

- Allows the detection of routing loops

The BGP minimizes the occurrence of routing loops. In addition to network reachability information, the BGP Protocol requires that update messages contain a list of the ASs the routing information has traversed. Routing loops are eliminated because a router never selects a path that contains its own AS.

- Selects routes based on performance and policy constraints.

The BGP allows a default weight to be added to all internal and external routes before computing the degree of preference for a route. If there are multiple routes to the destination networks, the route with the highest weight is chosen, allowing some routes to have higher priority than others.

The BGP allows user-configured AS-path policies and network number policies to be implemented. These policies determine whether a BGP speaker accepts and distributes routing information based on an AS-PATH attribute or IP network number.

External and Internal Peers

Two routers that exchange routing information using BGP are *peers*. Two kinds of peers exist:

- Internal peers

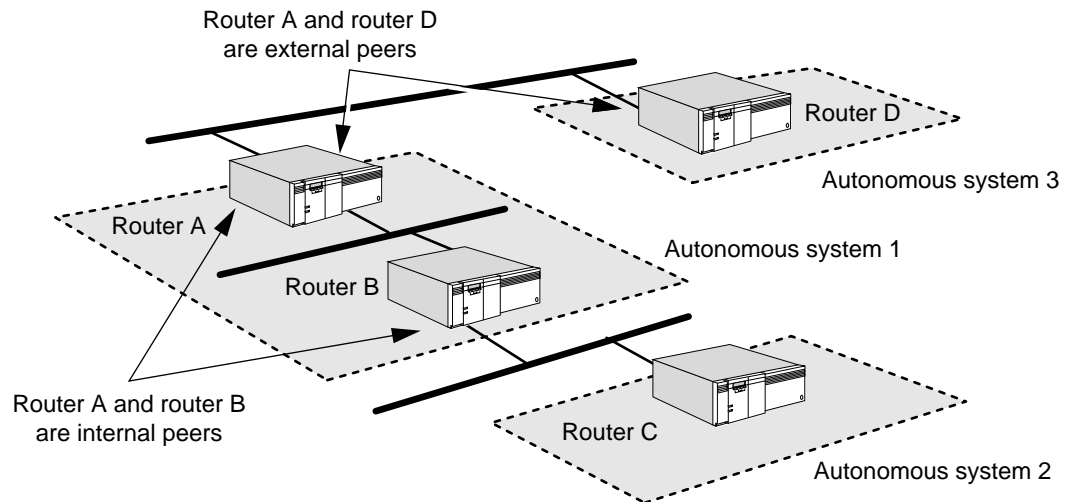
Two routers residing in the same AS are internal peers. Internal peers do not need to be attached to the same network.

- External peers

Two routers residing in adjacent ASs are external peers. External peers must be attached to the same network. BGP uses the common network to exchange messages between external peers.

See Figure 74 as an example. In this figure, routers A and B are configured as internal peers. Routers B and C are configured as external peers.

Figure 74 Internal and External Peers in an Autonomous System



Each peer establishes a BGP connection with the other peer. After the connection is established, the peers exchange update packets that indicate the networks each peer can reach. A peer also may report the networks that other gateways in other ASs can reach.

For example, in Figure 74, router B reports to router C all the networks that are reachable within autonomous system 1, and router C does the same for autonomous system 2. In addition to reporting the networks reachable within autonomous system 1, router B also reports to router C all the networks that are reachable through router A.

Peer-to-Peer Communication

After BGP peers are configured, three peer-to-peer communication states can be established between two peers:

- Connection establishment state
- Confirm state
- Established state

The router enters the connection establishment state immediately after you configure BGP and set up peers. In this state, Router B tries to establish a TCP connection with a configured gateway (Router C).

After a TCP connection is established, the routers exchange open messages in which the following information is exchanged:

- The version of BGP that a router wants to “speak”
- The hold time (maximum time for which a connection is kept open without receiving any keepalive or update packets)
- The AS number
- The Router ID

If the open messages are satisfactory, each peer enters the confirm state in which they exchange keepalive packets. When keepalive packets are received, the peers reach the established state, in which they exchange routing information in update messages.

The connection between peers is assumed to be a reliable TCP connection. Once the peers have exchanged routing tables, the only packets regularly exchanged (every 30 seconds) are keepalive packets. Routing updates occur only when new routes are reachable or previously advertised routes have become unreachable, which greatly reduces the amount of routing traffic and the time required to exchange and process information.

Update messages contain all reachable network addresses and the corresponding distances associated with each gateway, as well as the complete AS path for each network. Update messages may also contain explicit unreachable routes.

Path Attributes

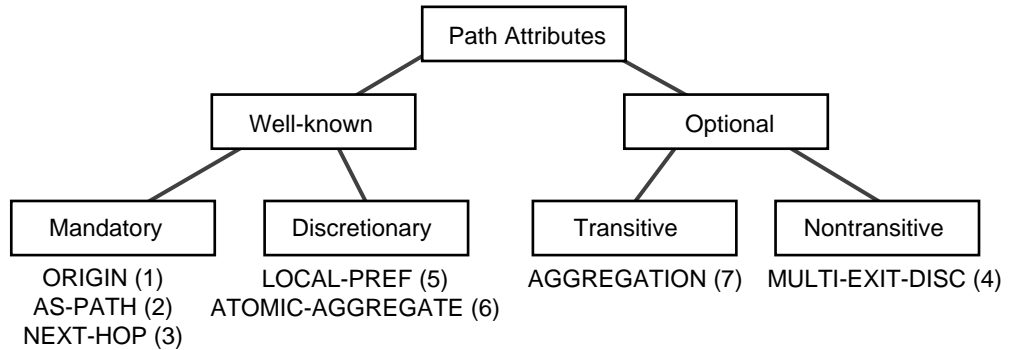
For each route, the BGP uses a set of path attributes to describe the route. These attributes help to eliminate looping of routing information, assist with policy-based routing decisions, indicate the original source of the path information as well as the IP address of the next-hop router to the destination, provide routing metrics, simplify the route selection process, and perform route aggregation.

Path attributes are classified as either well-known or optional as shown in Figure 75:

- A well-known attribute must be recognized by all BGP implementations.
 - The well-known attributes are further divided into mandatory and discretionary. A well-known, mandatory attribute (ORIGIN, AS-PATH, NEXT-HOP) must be included in every route description. A well-known, discretionary attribute (LOCAL-PREF, ATOMIC-AGGREGATE) may or may not be included in a route description depending on whether the attribute is implemented by the BGP speaker. A BGP speaker that receives a well-known path attribute is required to forward the attribute to its peer in update messages.
- An optional attribute may or may not be recognized by a BGP implementation.
 - The optional attributes are divided into transitive and nontransitive. An optional transitive attribute (AGGREGATION) may be passed along unchanged by a BGP router that has not implemented the attribute. An optional

nontransitive attribute (MULTI-EXIT-DISC) may not be passed along by a BGP router that has not implemented the attribute.

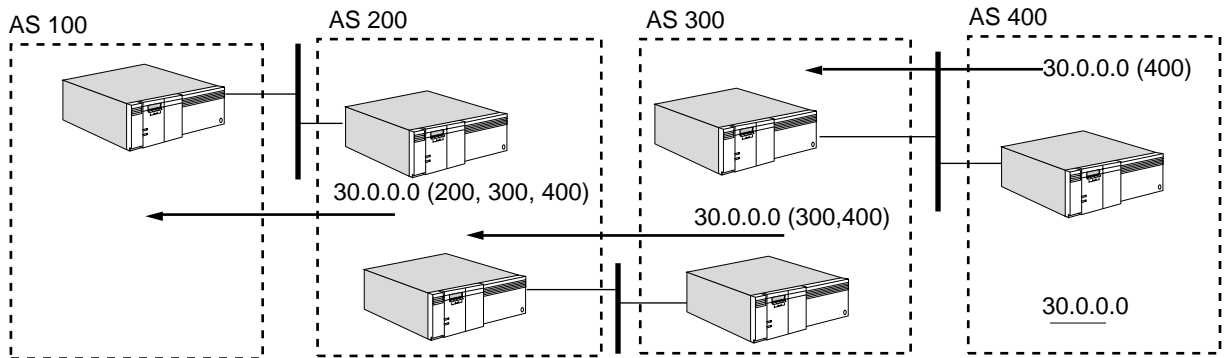
Figure 75 Path Attributes



You can display detailed information about path attributes associated with AS paths by entering the SHow -BGP ASPath Debug command.

AS-PATH The BGP Protocol uses the AS-PATH attribute to eliminate the occurrence of routing loops. As reachability information for a network traverses the internetwork, BGP creates a list of the ASs through which the routing information has passed. Each BGP speaker adds its own AS to the list before advertising network reachability to a peer as shown in Figure 76. The list of the ASs along the path to a destination network is called the AS-PATH attribute.

Figure 76 AS Path Example



The AS-PATH attribute is composed of a sequence of AS path segments. Each AS path segment may be either an AS SEQUENCE or an AS SET:

AS SEQUENCE An *ordered* set of ASs that the route in the update message has traversed.

AS SET An *unordered* set of ASs that the route in the update message has traversed. AS SETs are used by the route aggregation algorithm to reduce the size of the AS path information. An AS SET lists each AS number only once, regardless of how many times it may have appeared in the multiple AS paths that were aggregated.

An AS SET indicates that the destinations can be reached through paths that traverse at least some of the listed autonomous systems. AS SETs provide enough information to eliminate the looping of routing information.

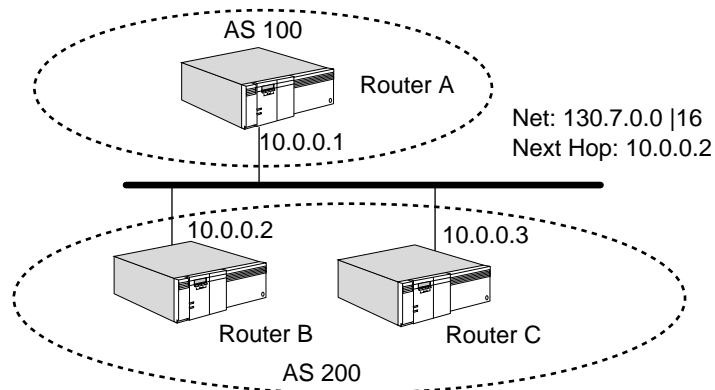
The AS-PATH attribute helps suppress routing loops. A router never accepts a route with its own AS in the AS-PATH list. The AS-PATH attribute can be used to make policy-based routing decisions. For more information, see “AS-Path-Based Policies” later in this chapter.

ORIGIN The ORIGIN path attribute defines the original source of the path information. The ORIGIN path attribute may contain the values IGP or Incomplete. IGP indicates that the destination network was learned by the original BGP speaker from the Interior Gateway Protocol (IGP) running in the original AS; this routing information is considered to be trustworthy.

Incomplete indicates that the routing information was obtained from some means other than an IGP. For example, the route may have been learned using a static configuration.

NEXT-HOP The NEXT-HOP path attribute defines the IP address of the border router that should be used as the next-hop to the destination networks. A BGP speaker can use its own IP address or the IP address of another router attached to the same subnet. This attribute allows a BGP speaker to advertise routes through another border router attached to the same subnet. For example in Figure 77, routers B and C are both border routers for AS 200. However, only router C is a BGP speaker and has a session with router A. Router C advertises the route to network 130.7.0.0 with router B as the next-hop router.

Figure 77 Next-Hop Router



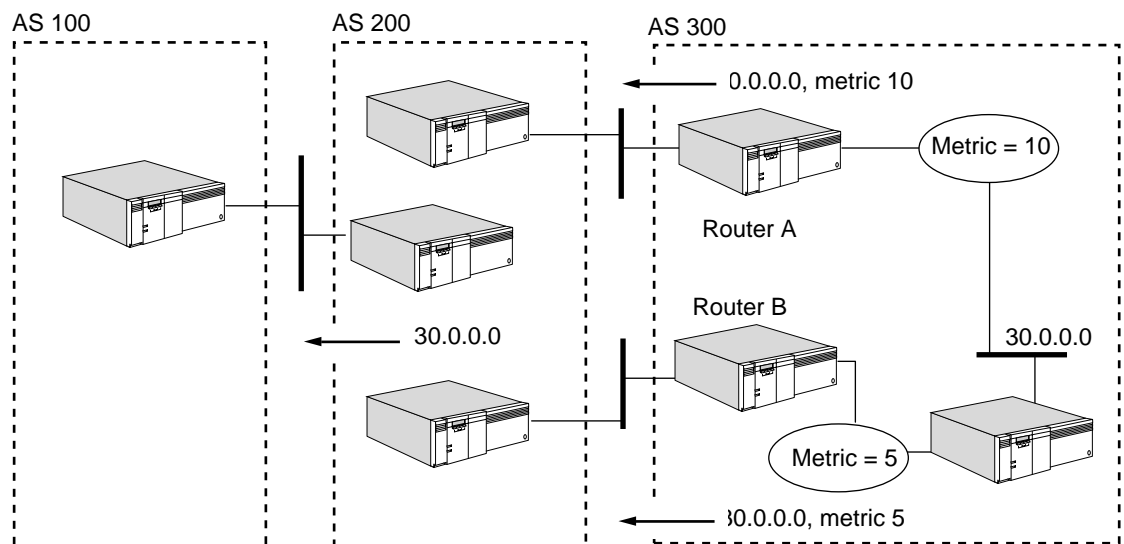
MULTI-EXIT-DISC The MULTI-EXIT-DISC attribute provides metric support. It has limited function in the BGP Protocol because routing loops are suppressed using the AS-PATH attribute instead of metrics. As routing information traverses multiple ASs, the ability to select a route based upon the least cost metric is no longer possible. One AS may use hop count, another uses a delay, and a third uses an administratively defined cost. Because no universally accepted metric is used, direct comparison of the combined metrics for routes using different paths has no real meaning.

The MULTI-EXIT-DISC attribute allows a BGP speaker to advertise a metric along with a route only if the network is internal to the same AS as the BGP speaker. If an AS has multiple BGP speakers to a neighboring AS, different speakers may advertise the same network with a different metric. Typically, BGP speakers select their metric based on the inter-AS cost to reach the destination network.

If the border routers of an AS receive different metrics for the same network, they compare the different metrics. The result of the comparison along with other factors determines the “best” route. If the MULTI-EXIT-DISC attribute is received over an external link, it may be propagated over internal links to other BGP speakers. However, it is never propagated to other BGP speakers in neighboring ASs.

In Figure 78, routers A and B advertise a route to network 30.0.0.0, which is completely contained within AS 300. Router A advertises the route with a MULTI-EXIT-DISC of 10. Router B advertises the route with a MULTI-EXIT-DISC of 5. The border routers in AS 200 make a comparison between the two metrics to select a better entry point in AS 300. However, the BGP speakers in AS 200 never propagate this metric to AS 100.

Figure 78 BGP Routing Metric



LOCAL-PREF The LOCAL-PREF attribute simplifies the route selection process. It advertises a degree of preference for each external route to BGP peers in the same AS so that a route with a higher degree of preference is selected over a route with a lower degree of preference.

This attribute is included as part of all update messages sent to other BGP speakers located within the same AS and never advertised to BGP peers in an adjacent AS.

ATOMIC-AGGREGATE The ATOMIC-AGGREGATE attribute is attached to a less specific route before propagating it to other BGP speakers to ensure that the aggregate is not deaggregated by other BGP speakers.

If a BGP speaker is presented with a set of overlapping routes from one of its peers, the more specific route takes precedence. If the BGP speaker selects the less specific route, the router attaches the ATOMIC-AGGREGATE attribute.

AGGREGATION The AGGREGATION attribute allows a BGP speaker performing route aggregation to advertise the AS that performed the aggregation. Aggregation is the process of combining several different routes so that a single route with a shorter mask can be advertised.

Path Selection

One of the most important tasks of BGP is to select the best path to a destination network based on the AS topology. In traditional routing protocols, each path has only a single metric to represent its cost. To evaluate two paths, the router compares the two metrics and selects the path with the lowest cost metric.

In interdomain routing, no universally agreed-upon metric among ASs can be used to evaluate different paths to a network. Therefore, each AS may implement its own set of criteria for path selection.

Path selection for the 3Com BGP-4 implementation is based on the following criteria in order of priority:

User-defined policies	Policies that are configured to control the distribution of routing information affect the paths that are available and the path selection process.
AS weight factor	When multiple paths exist to a given network, you can assign weights for AS paths or subsets of AS paths. The weight for a path is calculated by summing all the individual AS-path weight expressions that are applicable for the path. The path having the highest weight is selected.
AS count	If competing paths have the same weight, the path with a lowest total AS count is preferred over paths that have a larger AS count. BGP considers the path with a lower number of AS hops to be shorter.
link type	If competing paths have the same AS count, BGP prefers paths that arrive over external links to those that arrive over internal links.
path origin	If competing paths arrive over the same type of link, paths that are originated by an IGP Protocol has precedence.
MULTI-EXIT-DISC	If competing paths have the same ORIGIN attribute, BGP selects the path with the lowest MULTI-EXIT-DISC attribute value.
BGP-ID	If all of the above result in a tie, the selection process gives preference to the path from the peer with the larger IP address (BGP-ID). The IP addresses are compared as unsigned 32-bit integers.

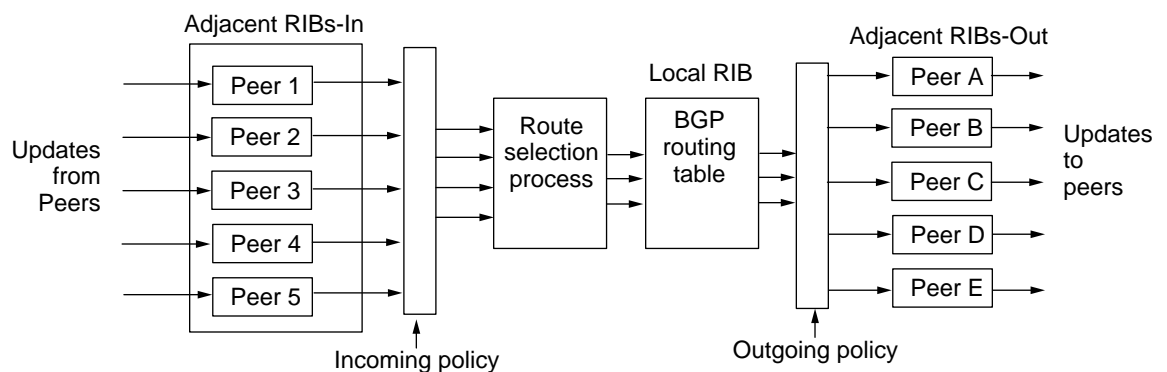
Policies

Policies determine whether AS-level routing information is accepted and distributed by a BGP speaker. Policies control routing information in two ways:

- Routing information can be received by a BGP speaker but not added to the routing table.
- Routing information can be received and added to the routing table but only advertised to some of the router's BGP peers.

BGP stores its routes in a routing information database (RIB), which is conceptually divided into three distinct parts as shown in Figure 79.

Figure 79 BGP Routing Selection and Policies



The Adjacent RIBs-In contains unprocessed routing information that has been received by the local BGP speaker from its peers. The information is learned from inbound update messages and represents routes that are available for input to the route-selection process of the local BGP speaker.

The Local RIB contains routes selected for use by the route-selection process of the local BGP speaker.

The Adjacent RIBs-Out organizes the routes that the local BGP speaker has selected for advertisement to its neighboring BGP speakers using outbound update messages.

Incoming policies are applied as part of the path selection process to manage the flow of information from the Adjacent RIBs-In to the Local-RIB. Outgoing policies are applied to manage the flow of information from the Local-RIB to the Adjacent RIBs-Out. The redistribution of routes is performed only on routes that have been placed in the Adjacent RIBs-Out.

The following list describes the three important effects of this sequence of operations:

- Incoming filters are implicitly applied before re-advertisement takes place.
- Only those routes used by the local BGP speaker are considered for re-advertisement.
- Only one outgoing route is advertised for each network even if many incoming paths for that route are learned from a variety of peers.

Interior Policies BGP learns network reachability information from many sources:

- Internal or external BGP speakers
- IGP speaker (RIP, OSPF, IISIS) residing in the same router
- Static route configurations
- Directly attached networks

To control the flow of routing information into BGP, an interior policy can be configured using the `InteriorPolicy` parameter. This parameter controls the blocking of IGP routes (RIP, OSPF, IISIS), static routes, and directly connected networks into BGP for advertisement to BGP speakers residing in adjacent autonomous systems.

By default BGP does not import any of the IGP, directly connected, or static routes.

Exterior Policies Each IGP Protocol controls the import of BGP routes through the configuration of its `ExteriorPolicy` parameter (in the RIP, OSPF, and IISIS Services).

BGP aggregated routes can only be leaked into IGP domains if the IGP routing protocol supports a mask along with each network route. OSPF and IISIS support this feature, but RIP does not provide this information in update packets. You cannot export aggregated routes into domains that run RIP.

Network Number-Based Policies Network number policies provide filtering of incoming and outgoing BGP advertisements based on IP network numbers. The following types of network filters can be configured:

- Do not accept routes for network `x.x.x.x` from BGP peer A.
- Do not advertise a route for network `z.z.z.z` to BGP speaker B.

The network number specified in these examples can be a single network number or a range of network numbers specified by a CIDR address prefix.

Network number policies are configured using the `NetworkFilter`, `NetPolicyAll`, `NetPolicyExt`, `NetPolicyInt`, and `NetPolicyPeer` parameters. The `NetPolicy` parameters allow you to configure the policy on all peers, external peers, internal peers, or on a specific peer.

AS-Path-Based Policies The AS-path policy provides filtering based on information contained in the AS-PATH attribute in each update message. Typical policies contain a combination of the following elements:

- Source AS
- Destination AS
- AS presence (within the AS-PATH attribute)
- Advertise or receive

Using these elements, the following policies can be configured:

- Distribute routes from AS 2 only to ASs 3, 6, and 7.
- Accept only those routes from AS 4 that have AS 7 contained in the path.

- Do not accept routes requiring a path through AS 3 from AS 10.
- Distribute routes containing AS 5 only to ASs 3 and 4.

To maintain consistent routing information within an AS, do not apply accept or deny policies to internal BGP peers.

AS-path policies are configured using the `AsFilter`, `AsPolicyAll`, `AsPolicyExt`, `AsPolicyInt`, and `AsPolicyPeer` parameters.

For more information about the AS-PATH attribute, see “AS-PATH” earlier in this chapter.

Route Aggregation

BGP route aggregation uses the Classless InterDomain Routing (CIDR) route aggregation strategy to combine several different routes so that a single route with a shorter mask can be advertised. By combining several networks into one supernet, the number of BGP messages sent to peers and the size of the routing table is reduced. Unnecessary details about subnets are hidden from peers. CIDR is a method of using IP addresses without regard to traditional address classes that helps reduce routing table growth by summarizing several networks or subnets with a single routing update.



Supernetting can only be understood and supported by protocols that carry mask information along with routes, such as OSPF and IISIS. The RIP Protocol always interpret routes as Class A, B, or C and cannot fully interpret routes with supernet masks. Do not configure BGP route aggregation in this situation.

BGP routers learn all the subnet routes through an intradomain routing protocol, such as OSPF, or static configuration. The BGP router may advertise to its BGP peers a single aggregate route that describes all the destinations connected to it. When a BGP router performs route aggregation, it needs to know the range of block of IP addresses to be aggregated or not aggregated.

The BGP router should aggregate as many routes as possible except those that cannot be treated as part of a single unit due to multi-homing, policies, or other constraints.

Aggregation should never encompass Class D address space (224.0.0.0 through 239.255.255.255).

Address Resolution

To resolve Internet addresses with associated Ethernet addresses when routing, the router uses the Address Resolution Protocol (ARP) as described in RFC 826.

Configure the `-ARP CONTROL` parameter to decide whether the router supports proxy ARP requests on the specified interface. A proxy request is a request for a target Internet address that is not on the subnet where the request originated. If the router generates proxy replies, it replies with its own Ethernet address, provided that it has a route in the routing table for the target subnet.

ARP determines the destination's Ethernet header format to be used by sending out ARP requests that include the format. Specify the format by configuring the `RequestFormat` parameter in the ARP Service. The system replying to the request then uses the Ethernet header format it supports, and the router records the IP

address, the Ethernet address, and the Ethernet header format of the replying system. The information is valid for the time specified by the HoldTime parameter in the ARP Service. Configure the time if you want the router to hold the information for more than or less than 24 hours, which is HoldTime's default value.

Inverse ARP

Inverse ARP is an adaptation of ARP that resolves DLCIs on Frame Relay networks to IP addresses, as described in RFC 1293.

Extended ARP

Extended ARP is an adaptation of ARP that resolves internet addresses to E.164 addresses on SMDS networks, as described in RFC 1209.

Other Global Router Configurations

After you determine how the router should route packets on each of its interfaces, you can influence the global router operation in several areas, as follows:

- Treatment of Internet Control Message Protocol (ICMP) request packets and generation of packets

Configure the -IP ICMPReply parameter to control whether the router responds to Address Mask Request and Information Request packets. The router supports all the ICMP messages described in RFC 1009.

Configure the -IP ICMPGenerate parameter to control whether the router originates ICMP ReDirect, Destination Unreachable, and TimeExceed packets.

- How long the IP layer waits for all IP fragments

Configure the -IP ReassemblyTime parameter to control the length of time the IP layer waits for all IP fragments of an IP datagram to be received. This parameter applies only to packets specifically destined for the local router.

- The level of security

Use the parameters beginning with "Sec" in the IP Service to configure the system for IP security options processing.

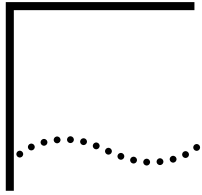
- The value of time-to-live (TTL)

Configure the -IP DefaultTTL parameter to specify the value the router puts in the TTL field of an IP packet when it generates the packet.

- Prioritization of packets within the IP Protocol

Some actions of the -IP FilterAddrs parameter enable your bridge/router to do special processing of IP packets over WAN links, improving the IP WAN traffic management. When the FilterAddrs parameter is used with the -IP Filters parameter, you can specify the packets to which the special processing applies.

For more information on the parameters discussed in this section, see the IP Service Parameters chapter in *Reference for Enterprise OS Software*.



CONFIGURING QUALITY OF SERVICE

This chapter describes how to configure Quality of Service (QoS) policies. This chapter provides a conceptual overview of QoS and gives guidelines for operation.

For conceptual information, see “About Policy-Based QoS Management” later in this chapter.

Configuring IP Packet Classification Services

IP policy rules generally require packet classification to identify packets that satisfy the specified matching criteria before the policy action can be applied. For this reason, IP provides a packet classification service that can be used by other IP-based modules to configure flow-based policies.

Configuring a Classifier List

The ClassifierList parameter in the IP Service is used to define a list of flow-based packet classification criteria based on the fields in the IP packet headers (IP/UDP/TCP).

A classifier list is used to classify IP traffic into an aggregated flow (for example, all packets from a specific IP subnet) or specific application flow between two end systems (for example, FTP sessions between two hosts) using the following:

- Source/destination IP address
- Protocol type
- Source/destination UDP/TCP port
- TOS byte value

A classifier list cannot be deleted until all active policies associated with this classifier list are deleted.

To create a classifier list use:

```
ADD -IP ClassifierList <name>(1-15 char) <priority> (1-9999)
  [Exclude]
  [From (<ipaddr> [/<prefix>(0-32)]) | <ipaddr_list_name>]
  [To (<ipaddr> [/<prefix>(0-32)]) | <ipaddr_list_name>]
  [TOS <phb> | <value> (hex 0-FF) [ / <mask> (hex 0-FF) ]
  (ServiceList <service_list_name> |
  (PROTOCOL <protocol>(0-255)) |
  (TCP [Src <compare> <port> | <port1-port2> ]
    [Dst <compare> <port> | <port1-port2>]
  (UDP [ Src <compare> <port> | <port1-port2>]
    [Dst <compare> <port> | <port1-port2>]
  (ARCHIE | BGP | ICMP | DNS | FINGER | FTP | GOPHER | GRE |
  HTTP | NFS | NNTP | NTP | OSPF | POP | RIP |
  RPLAYER | RSVP | SECHTTP | SMTP | SNMP | SOCKS | SYSLOG |
```

TELNET | WAIS | WHOIS)

where:

- <name>: Unique name of up to 15-characters.
- <priority>: 1-9999, 1 = highest. If multiple classifiers exist in a classifier list, the priority determines the precedence order in the matching operation. The first match terminates the search.
- Exclude: Packets matching the specified criteria are excluded.
- <prefix>: IP prefix length (0-32).
- <ipaddr_list_name>: name of IP Address List created via ADD -IP AddressList
- <protocol>: Protocol number 0-255.
- <phb>: IETF-defined behavior. See RFC 2474 for details.
EF1|EF2|AF1|AF2|AF3|AF4|P1|P2|P3|P4|P5|P6|P7

where:

- EF_n denotes the IETF differentiated Service Explicit Forwarding Per Hop Behavior.
- AF_n denotes the IETF differentiated Services Assured Forwarding Per Hop Behavior group n.
- P_n denotes the IP Precedence Per Hop Behavior group n.
- <value>: 0-0xff; TOS byte value. If the optional <mask> is specified, the TOS byte in the packet's IP header is ANDed with <mask> and the result is compared against <value>. For example, if the differentiated services codepoint (bits 0-5) match is desired, mask 0xfc is used; for IP-precedence (bits 0-2) match, mask 0xe0 should be used.
- <compare>: > (greater than), < (less than) or = (equal).
- <port>: TCP or UDP port (0-65535).
- <service_list_name>: name of IP service list created using the ADD -IP ServiceList parameter.

For example, to create classifier lists named CL_1 and CL_2, enter:

```
ADD -IP ClassifierList CL_1 10 from AL_1 to AL_2 TOS P5
ADD -IP ClassifierList CL_1 20 from 10.0.0.8 to 20.0.0.8 SVL SL_1
ADD -IP ClassifierList CL_2 10 to 30.0.0.0/24 TCP
```

To display the classifier lists, enter:

```
SHow -IP ClassifierList
```

```
-----IP Classifier List -----
```

```
Name: CL_2 Users:0
```

```
Priority: 10 To: 30.0.0.0/24 TCP
```

```
Name: CL_1 Users:0
```

```
Priority: 10 From AL_1 To: AL_2 TOS:P5
```

```
Priority: 20 From: 10.0.0.8/32 To:20.0.0.8/32 SerViceList: SL_1
```

Configuring an Address List

The address list is a group of IP address/mask entries that can be referenced using a unique AddressListName. To create an address list, use:

```
ADD -IP AddressList <name>(1-15 characters) <IP address>[/<mask>(0-32)]
```

For example, to configure an address list named AL_1, enter:

```
ADD -IP AddressList AL_1 10.0.0.1
ADD -IP AddressList AL_1 20.0.0.1
ADD -IP AddressList AL_1 129.0.0.0/8
```

To display the contents of the address list AL_1, enter:

```
SHoW -IP AddressList AL_1
----IP Address List ---
Name: AL_1 Users:0
10.0.0.1/32
20.0.0.1/32
129.0.0.0/8
```

Configuring a Service List

The service list is one or more IP service entries that can be referenced by a unique service list name. To create a service list, use:

```
ADD -IP SerViceList <name>(1-15 char)
(PROTocol <protocol>(0-255)|
(TCP ([Src <compare> <port> | <port1-port2>]
(Dst <compare> <port>|<port1-port2>]))|
(UDP ([Src <compare> <port> | <port1-port2>]
(Dst <compare> <port>|<port1-port2>]))|
(ARCHIE | BGP | ICMP | DNS | FINGER | FTP | GOPHER | GRE |
HTTP | NFS | NNTP | NTP | OSPF | POP | RIP |
RPLAYER | RSVP | SECHTTP | SMTP | SNMP | SOCKS | SYSLOG |
TELNET | WAIS | WHOIS)
```

For example, to create a service list named SVL_1, enter:

```
ADD -IP SerViceList SVL_1 TCP
ADD -IP SerViceList SVL_1 UDP
ADD -IP SerViceList SVL_1 GRE
```

To display the IP Service list contents, enter:

```
SHoW -IP SerViceList SVL_1
-----IP Service List -----
Name: SVL_1 Users:0
TCP
UDP
GRE
```

Configuring IPQoS

This section describes using the QoS service parameters.

POLicy Parameter

The QoS POLicy parameter is a per-port parameter. You can define any number of QoS policies on a port. Inbound policies are applicable on packets received on the ingress ports. Outbound policies are applicable on packets transmitted over the egress ports. To add a policy, use:

```

ADD !<port> -IPQoS POLicy <policy_name> <priority> <cl_name> < direction >
  [RateLimit <avg rate> <peak rate> <burst >]
  [ExcessAction Transmit | Drop | SetTOS <tos>[/<mask>]]
  [ConformAction Transmit | Drop | SetTOS <tos>[/<mask>]]
  [(ClassBasedQue <class_name>) | ProtocolRsrv <name_tag>) |
    (PriorityQue Urgent | High | Medium | Low)]
  [VlanPriority < 802.1P >]
  [NextHop (<ipaddr>[<ipaddr>[<ipaddr>]])|(!<port>[!<port>[!<port>]])]

```

where:

- <policy_name>: Unique name (1-15 characters) for this policy.
- <cl_name>: Unique name of the classifier list containing one or more packet classification rules; each defines a set of packet-matching criteria. If the reserved name NULL is used, no packet matching criteria is specified, the policy rule matches all packets and thus becomes the default policy on the specified port.
- <direction>: INbound | OUTbound; specifies whether this policy applies to inbound or outbound traffic.
- <priority>: 1 - 9999, 1 = highest priority; specifies the precedence order of this policy in the packet matching operation.
- <tos>: 0-0xff; set the TOS byte in IP header to the specified hexadecimal value.
- <mask>: 0-0xff; if the optional mask is specified, 0-bits in the mask specify the bit locations in the TOS-byte that must remain unchanged while 1-bits specify those in the TOS-byte that must take on the corresponding bits in the specified TOS value.
- <class_name>: CBQ class name (1-15 characters).
- <name_tag>: Protocol Reservation name tag.

IPQoS does not police or mark packets unless an explicit QoS policy is defined for the packet flow. If there is no policy match, the default action is to assign the packet to the default service class.

QoS Action Attributes

The following sections describe the QoS policy actions that can be applied to a packet that satisfies the packet-matching criteria defined in the associated classifier list.

RateLimit RateLimit <avg rate> <peak rate> <burst>

Rate limiting can be specified on the inbound and outbound port. Depending on the packet rate, packet classification and rate limiter may not scale on the high-speed LAN interfaces.

Traffic is metered based on the specified token bucket parameters.

- <avg rate> specifies the average transfer rate in kbps.
- <peak rate> specifies the peak rate in kbps.
- <burst> specifies the maximum burst size in bytes.

The default is no rate limiting.

Excess Action ExcessAction Transmit | Drop | SetTOS <value>[/<mask>]

This attribute specifies the handling of packets that exceed the specified peak rate:

- Transmit: Transmit the packet.
- Drop: Discard the packet.
- SetTOS: Mark TOS byte to the specified value and transmit the packet. If the optional mask is specified, the 0-bits in the mask specify bits in the TOS byte that must not be changed and the 1-bits specify those that must take on the corresponding bits in the specified TOS value. For example new TOS byte = (mask & old TOS byte) | (mask & value), where mask is the bitwise complement of mask.

The default is Drop.

Conform Action ConformAction Transmit | Drop | SetTOS <value>[/<mask>]

This attribute specifies the handling of packets that conform to the traffic rate. If RateLimit is not specified, this attribute specifies handling of packets that match the classification criteria:

- Transmit: Transmit the packet.
- Drop: Discard the packet.
- SetTOS: Mark TOS byte to the specified value and transmit the packet. If the optional mask is specified, the 0-bits in the mask specify bits in the TOS byte that must not be changed and the 1-bits specify those that must take on the corresponding bits in the specified TOS value, for example new TOS byte = (mask & old TOS byte) | (mask & value), where mask is the bitwise complement of mask.

The default is Transmit.

ClassBasedQue ClassBasedQue <class_name>(1-15 characters)

The ClassBasedQue attribute designates the CBQ class for the packet flow. The action can be applied only at the egress port.

The egress port must be configured to run the CBQ queueing policy; <class_name> must be a known CBQ class that will provide the desired service level for the packet. If the named class does not exist, the default CBQ class on the port will be assigned instead.

If ClassBasedQue is not specified, the Differentiated Service Code Point (DSCP) value in TOS-byte determines the CBQ class assigned to the outbound packet if the TOS to CBQ class mapping is enabled on the egress port using the TOSMap parameter. An unrecognized DSCP value maps to the default CBQ class.

ProtocolRsrv Protocol Rsrv <name_tag>

The ProtocolRsrv attribute designates a defined protocol reservation name tag for the packet flow. The action can only be applied at the egress port, which must be configured to run the protocol reservation queueing policy.

PriorityQue PriorityQue Urgent | High | Medium | Low

The PriorityQue attribute designates a priority for the packet flow on the egress port, which must be configured to run the priority queue queuing policy.

VLANPriority VLANPriority <802.1P> (0-7)

VLANPriority attribute assigns a specific IEEE 802.1P priority value to an outbound VLAN packet.

If the packet is not tagged with an 802.1P priority value, the IP/TOS-byte value determines the IEEE 802.1P priority value in the 802.1Q tag if the OutVlanMap parameter is enabled on the egress VLAN port.

NextHop This attribute redirects traffic to the specified nexthop or the specified egress port. The action can be applied only at the ingress port.

InVlanMap Per-port Parameter

The InVlanMap parameter defines the non-default mapping of IEEE 802.1P priority to IP/TOS value for inbound IP traffic at the specified VLAN port. Default mapping is mapping the 3-bit 802.1P value directly into the 3-bit IP-Precedence fields in the TOS_byte with the remaining TOS bits set to zero. To set the inbound VLAN Mapping use:

```
SETD !<port> -IPQOS InVlanMap = Default | (<tos_0>... <tos_n>)
```

For example, to set and display inbound VLAN mapping, enter:

```
SETDefault !v33 -IPQOS InVlanMap=00 20 40 40 40 a0 a0 e0
SETDefault !v33 -IPQOS CONT=IVM
SHow !v33 -IPQOS InVlanMap
-----Port !V33 802.1P to TOS Map ----
802.1P  TOS
-----  ---
      0   00
      1   20
      2   40
      3   40
      4   40
      5   A0
      6   A0
      7   A0
```

OutVlanMap Per-port Parameter

This parameter defines the mapping of IP/TOS-byte to IEEE 802.1P priority for outbound IP traffic at the specified VLAN port. If the TOS to 802.1P mapping is enabled via the CONTrol parameter, one or more TOS-byte values can be mapped into a 802.1P priority value. The mapping is applied after policy filtering only if the packet is not already marked with a valid IEEE 802.1P priority value. To set the outbound VLAN map use:

```
ADD !<port> -IPQOS OutVlanMap <vlan_priority>
      Default | ([Mask <mask>] <tos>... <tos_n>)
```

To add and display an outbound VLAN map, enter:

```
ADD !v33 -IPQOS OutVlanMap 5 default
```



```
ADD !v33 -IPQOS OutVlanMap 1 20 28 2a
SHOW !v33 -IPQOS OutVlanMap
```

```
----- Port !V33 TOS to 802.1_ Map ---
TOS      802.1P
---      -----
20       1
28       1
2A       1
**       5 (default)
```

TOSMap Per-port Parameter

If TOS to CBQ class mapping is enabled via the **CONTROL** parameter, one or more TOS-byte values can be mapped into a CBQ class. Mapping is applied on an outbound packet only if a CBQ class has not already been assigned via an IPQoS policy. To add a TOS to CBQ class mapping, use:

```
ADD !<port> -IPQOS TosMap <class_name>
      Default | ([Mask <mask>] <tos>... <tos_n>)
```

For example, to add a TOS to CBQ class mapping, enter:

```
ADD !7 -IPQOS TosMap CBQ_EF1 EF1 # map DSCP-EF to CBQ class CBQ_EF1
ADD !7 -IPQOS TosMap CBQ_AF1 AF1 # map DSCP-AF1 to CBQ class CBQ_AF1
ADD !7 -IPQOS TosMap CBQ_AF2 AF2
ADD !7 -IPQOS TosMap CBQ_AF3 AF3
ADD !7 -IPQOS TosMap CBQ_AF4 AF4
```

CONTROL Per-port Parameter

Per port control of QoS filtering is allowed. To control QoS functions on a port use:

```
SETDefault !<port> -IPQOS CONTROL = (
      InFilter | NoInFilter, OutFilter | NoOutFilter,
      InVlanMap | NoInVlanMap, OutVlanMap | NoOutVlanMap,
      TosMap | NoTosMap)
```

For example, to enable inbound VLAN mapping on port !v33, enter:

```
SETDefault !v33 -IPQOS CONT=IVM
```

To enable inbound policy filtering on port !V33, enter:

```
SETDefault !V33 -IPQOS CONTROL=InFilter
```

About Policy-Based QoS Management

Policy-based QoS ensures that mission-critical and real-time application traffic get adequate network resources to traverse the network regardless of the competing demands for bandwidth by other applications.

The policy-based QoS management enables you to control bandwidth allocation and service levels on IP traffic flows. Traffic flows can be metered and policed to ensure its bandwidth consumption does not exceed the defined rate limits, and the conforming and excess traffic is handled per policy. When multiple flows are aggregated into a service class, rate limiting protects conforming flows from the aggressive flows hogging network resources that may lead to a denial of service. Flows can also be policed to ensure correct marking of the IP/TOS-byte in IP header as per policy.

Given the scalability problems associated with RSVP's per flow state and signalling overheads at every hop, the emerging IETF standard for scalable end-to-end QoS - IP Differentiated Service is supported. Incoming traffic flows can be classified into a small number of service classes per defined QoS policy and the bridge/router provides the service level that corresponds to the DSCP via a CBQ (Class Based Queue) packet scheduler and the RED (Random Early Detection) congestion avoidance mechanisms. Due to the per packet cost of CBQ/RED algorithms, these queue management policies are only supported over the FR/PPP WAN links.

Flexible QoS control is configured via the IPQoS Service as port-specific policies. QoS policies can be applied to the inbound traffic at the ingress port and/or the outbound traffic at the egress port. QoS policies are associated with flows. Policies are stored in the user-defined precedence order in the QoS policy database. The policy action associated with the first matching policy found for the packet will be applied. Flow can be defined as either an aggregated flow or a specific application flow between two end systems. Flows are classified via the packet classification service provided by IP.

QoS Policies The following types of QoS policy can be defined:

- **Bandwidth control:** If rate limiting is specified in a QoS policy, the associated traffic flow will be metered and policed. Rate limiting can be applied to traffic transmitted or received on an interface. You may also define actions, such as forward/discard/remark TOS-byte, to handle traffic that conforms to or exceeds the rate limit.
- **TOS control:** the IP/TOS byte can be set to a specified TOS value, which allows incoming packets to be classified into a small number of DSCP-based classes. The TOS byte can also be remarked for forwarding to another administration domain with a different IP/TOS conventions.
- **802.1P control:** a specific 802.1P priority value can be assigned to a flow at the egress VLAN port.
- **Service class control:** A specific service class (CBQ class name/Priority Queue/ Protocol Reservation tag) can be assigned to a flow independent of the DSCP value in the TOS byte.
- **Traffic redirect:** Traffic can be redirected at the ingress port to the user-defined nexthop.

IEEE 802.1P Prioritization If 802.1P to TOS mapping is enabled on the ingress VLAN port, the 802.1P priority in 802.1Q tag of the incoming IP packet will determine the IP/TOS value based on the default or user-configured mapping. This operation is applicable when the ingress port is connected to a VLAN network.

Likewise, the IP/TOS value will determine the 802.1P priority of the outgoing packet based on the default or user-configured mapping, if TOS to 802.1P mapping is enabled on the egress VLAN port. This operation is applicable when the egress port is connected to a VLAN network.

IP traffic can also be classified via a QoS policy to be tagged with a specific 802.1P priority.

Class-Based Queueing Configuration

In addition to the Priority Queueing and Protocol Reservation queueing policy, the following class-based queueing functions can be configured using the PORT Service.

- Class Based Queueing

Class Based Queueing (CBQ) is a link-sharing packet scheduler that combines priority queueing with weighted round-robin scheduling. It CBQ allows partitioning of link bandwidth by hierarchically structured classes. Packet scheduling is priority based, with higher priority class traffic transmitted ahead of lower priority class traffic, and traffic is weighted round-robin between classes with the same priority. Each class is regulated to its allocated bandwidth, and a child class can be optionally configured to borrow bandwidth from its parent class with excess bandwidth.

To minimize queue latency, CBQ is only supported on the Frame Relay and PPP physical ports. It is not allowed on a virtual port. All associated virtual port traffic is aggregated onto the CBQ classed configured on the physical port.

- CBQ Rate Limit

CBQ rate limiting can be configured on PPTP/L2TP/IPIP and Frame Relay virtual ports only. A token bucket rate limiter can be defined for the port traffic or one or more of the CBQ classed configured on the associated egress physical port. Packets in excess of the token bucket peak rate can either be discarded or transmitted downstream to the physical port with a high drop precedence. If RED is enabled on the target CBQ class, the excess packets with the higher drop precedence values are preferentially discarded to free up queue slots for packets that conform to the specified token bucket rate.

- Random Early Detection (RED)

RED can only be enabled on CBQ classes. RED manages CBQ class queue size by dropping packets probabilistically based on the estimated average queue size. The drop probability increases linearly as the average queue size grows. Packet drops begin when the average queue size exceeds the system-defined minimum threshold and all arriving packets are dropped at the maximum drop probability when the average queue size reaches the system-defined maximum threshold. All incoming packets are dropped when the absolute class queue limit is reached. The minimum and maximum thresholds are based on the class bandwidth and the packet drop precedence value as marked by the CBQ Rate Limit if rate limiting is enabled for the virtual port traffic. Packets marked green (at or below the average rate) have a higher threshold than yellow packets (exceeds the average rate but at or below the peak rate), which in turn maintains a higher threshold than the RED packets (exceeds the peak rate).

RED is most effective on TCP flows, which throttle back transmission rate when packet loss is detected.

Configuring Per-Port CBQ Rate Limit

Outbound virtual port traffic can be rate limited using:

```
ADD !<port> -PORT RateLimit "ROOT"
    <avg rate> <peak rate> <burst> [TransmitExcess]
```

If a rate limiter is defined for the "ROOT" class, all port traffic will be metered through the token bucket defined for the ROOT class.

Otherwise, CBQ classes configured on the egress physical port can also be rate limited using:

```
ADD !<port> -PORT RateLimit <class_name>
    <avg rate> <peak rate> <burst> [TransmitExcess] [Default]
```

where:

- <class_name>: CBQ class name; rate limit is applied to the traffic designated for the CBQ class. If a rate limiter is defined for the "DEFAULT" class, all virtual port traffic tagged for service classes with no configured rate limiter is metered through the token bucket defined for the DEFAULT class. The rate limit for the default class must be configured first.
- <avg rate>: average rate in kbps.
- <peak rate>: peak rate in kbps.
- <burst>: burst size in bytes.
- TransmitExcess: specifies that packets exceeding the rate limit are tagged as discard eligible. Discard eligible packets are transmitted only if the specified service class is not congested. This option is supported only if the random early discard (RED) is enabled on the CBQ class. If TransmitExcess is not specified, all excess traffic is discarded.
- Default: specifies the class is the default class defined for the egress physical port.

The default rate limit for a tunnel port is:

```
ADD !<port> -PORT RateLimit "ROOT" 64 128 4096
```

Rate limit is applied only if the QueueCONTROL parameter on the parent port is configured with CBQRateLimit.

For example, to configure and display rate limits for ports, enter:

```
ADD !v11 -Port RateLimit EF 8 8 1024
ADD !v30 -Port RateLimit root 64 64 2048 def
SHoW !v30 -Port RateLimit
-----CBQRateLimit on Port !V30-----
CBQ ClassName Avg Rate kbps PeakRate kbps Burst (bytes) ExcessAction
-----
*Root          64          64          2048          Drop
```

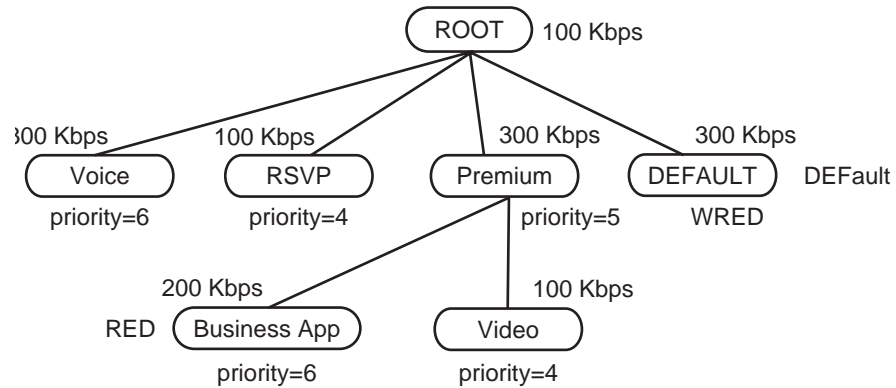
For another example of configuring and displaying rate limits for ports, enter:

```
ADD !v30 -Port RateLimit AF4 64 64 1024 TransmitExcess Def
ADD !v30 -Port RateLimit EF 8 8 1024
ADD !v30 -Port RateLimit AF1 32 64 1024 TransmitExcess
SHoW !v30 -Port RateLimit
-----CBQRateLimit on Port !V30-----
CBQ ClassName Avg Rate kbps PeakRate kbps Burst (bytes) ExcessAction
-----
AF1            32            64            1024          Xmit
*AF4           64            64            1024          Xmit
EF             8             8             1024          Drop
```

Configuring Class Based Queueing

A sample CBQ configuration with class hierarchy is shown in Figure 80.

Figure 80 Class Based Queueing Configuration Example



ClassBasedQue Parameter

CBQ classes are defined with the ClassBasedQue parameter under the PORT Service using:

```
ADD !<port> -Port ClassBasedQue <class_name> <parent_name> <bandwidth>
[AvgPacketSize <bytes>] [ Priority <value>] [ Default ][RED] [Borrow]
```

where:

- <class_name> is a 1-15 character name tag for this class. Must be unique within the CBQ class hierarchy for this port. Up to 512 classes can be defined. Following class names are reserved:
 - "ROOT" is reserved for the top-level root class.
 - "DEFAULT" is reserved for the default service class. Packet traffic that is not assigned a service class or an undefined service class is assigned to the default service class. Default class is removed if a user configured default service class is created.
 - "RSVP" is reserved for RSVP traffic only.
- <parent_name> is a 1-15 character name tag of the parent class for this class. A NULL is used if this class is the root class. Parent class must be defined ahead of its child classes.
- <bandwidth> is the bandwidth in kbps allocated to this class. The aggregate at each level of the class hierarchy should add up to 100% of the parent bandwidth. A multiple of 8 kbps should be used.
- RED enables Random Early Detection capability on this class queue. RED uses an exponentially weighted moving average estimator to compute the average queue size (which smooths out the bursty packet flow). The probability of packet drop increases as the average queue size increases. The selection of the mark probability drop probability determination is based on the link speed. The RED operational parameters is set to system-defined values.
- Borrow specifies whether borrowing from the parent class is allowed when the class exceeds its allocated bandwidth.

- AvgPacketSize is the average packet size in bytes to be used in computing the CBQ inter-packet interval (that is, AvgPacketSize/ClassAllocatedBandwidth), which determines if the class is over/under-limit by comparing against the current sending rate. Default size is MTU of the interface. Though CBQ parameters are designed not to be too sensitive to this packet size, setting appropriate average packet size is important for optimal CBQ performance. If the specified size is too large (vs. the real packet traffic), the class will not achieve its target rate while too small a value will result in the class exceeding its target rate.
- Priority specifies the class priority between 0 to 7. Default is 1. Higher priority classes are schedule ahead of lower priority classes.
- DEfault indicates the designated default class.

For example, to configure class based queueing, enter:

```
ADD !5 -Port ClassBasedQue root null 1024
ADD !5 -Port ClassBasedQue EF root 64 priority 7
ADD !5 -Port ClassBasedQue AF1 root 256 priority 5 red borrow
ADD !5 -Port ClassBasedQue AF2 root 256 priority 4 red borrow
ADD !5 -Port ClassBasedQue AF3 root 128 priority 3 red borrow
ADD !5 -Port ClassBasedQue AF4 root 128 priority 2 red borrow
ADD !5 -Port ClassBasedQue rsvp root 128 priority 5
SHoW !5 ClassBasedQueue
-----ClassBasedQue on Port !5-----
Name      Parent      Kbps      Priority      APS
-----
ROOT      NULL        1024      0             1500
AF4       ROOT        128       2             1500      RED BOR
AF3       ROOT        128       3             1500      RED BOR
AF2       ROOT        256       4             1500      RED BOR
AF1       ROOT        256       5             1500      RED BOR
EF        ROOT        64        7             1500
RSVP     ROOT        128       5             1500
DEFAULT  ROOT        64        6             11500     DEF
```

QueueStatistics Parameter

The QueueStatistics parameter displays the CBQ statistics. If RED is enabled for a CBQ class, the drop statistics are also included using:

```
SHoW [!<port>] -PORT QueueStatistics
```

The following CBQ class statistics are maintained:

- XmitPackets: The number of packets transmitted.
- XmitBytes: The number of bytes transmitted.
- DropPkts: The number of packets dropped (tail drop or by RED).
- Borrows: The number of (bandwidth) borrow operations executed.
- Delays: The number of timer delays initiated due to over-limit conditions.

Examples

To display CBQ statistics, enter:

SHow !5 -Port QueueStatistics

```
-----Port !5 QueueCONT=ClassBasedQue-----
ClassName      XmitPkts      XmitBytes      DropPkts      Borrows      Delays
-----
ROOT           0              0              0             0            0
AF$            0              0              0             0            0
AF3            0              0              0             0            0
AF2            0              0              0             0            0
AF1            0              0              0             0            0
EF             0              0              0             0            0
DEFAULT        55             1815           0             0            0
URG :Xmit =    O, Discard = 0,      InQ = 0
Controlled Delay = ms,      In Driver Queue = 0 bytes
```

The following CBQ Rate Limit statistics are maintained:

- GreenPkts: The number of packets transmitted at or below the token bucket average rate.
- YellowPkts: The number of packets transmitted that exceed the average rate but are at or below the token bucket peak rate.
- RedPkts: The number of packets that exceed the token bucket peak rate; if the Transmit Excess option is not specified for the associated CBQ class, this also indicates the number of packets dropped.

To display CBQ rate limit statistics, enter:

```
SETDefault !v30 QCONTROL = CBQRL
SHow !v30 -Port QueueStatistics
```

```
-----Port !V30 QueueCONT=CBQRateLimit-----
CBQ ClassName  GreenPkts      YellowPkts      RedPkts      ErrDiscards
-----
AF$            0              0              0            0
AF4            0              0              0            0
EF             0              0              0            0
```

QueueCONTROL Parameter

The QueueCONTROL parameter enables the CBQ queuing policy on a port using:

```
SETDefault !<port> -PORT QueueCONTROL = (PriorityQueues | PROTOcolRsrv |
ClassBasedQue | CBQRateLimit | None)
```

CBQRateLimit enables the CBQ rate limiters defined for the CBQ classes on the specified virtual port.

Example 1: A bridge/router with an 100 Mbps port 1 interface is sending too much traffic to ports 3 and 4, with subnets 151.0.0.0 and 152.0.0.0 respectively.

To configure example 1, follow these steps:

To limit the traffic from port 1 to ports 3 and 4, do the following operations.

- 1 Create a classifier list which will match packets for the subnets 151.0.0.0 and 152.0.0.0.

```
ADD -IP ClassifierList CL1 20 to 151.0.0.0/16
ADD -IP ClassifierList CL2 20 to 152.0.0.0/16
```

- 2 Define the policy on port 1, applying the rate limit action associated with the above defined classifier list. All packets with destination subnet 151.0.0.0 and 152.0.0.0 will be limited to an average of 128 kbps, at a peak rate of 256 kbps, and maximum burst of 3000 bytes.

```
ADD !3 -IPQos POLicy POL1 20 CL1 INbound
RateLimit 128 256 3000
ADD !3 -IPQos POLicy POL2 20 CL1 INbound
RateLimit 128 256 3000
```

- 3 Enable the inbound Quality of Service Policy filter with the following command:

```
SETD !3 -IPQos CONTrol=InFilter.
```

Example 2 This example shows rate limiting outbound traffic based on a protocol.

In this example, the amount of traffic being transmitted on port 3 is limited. The type of traffic which is limited, is defined by the type of protocols. FTP traffic is limited to a maximum rate of 256 kbps, and all HTTP traffic at 128 Kbps. All FTP and HTTP exceeding the defined rate are discarded.

To limit the outbound traffic on port 3 by protocols, follow these steps:

- 1 Create a classifier list which will match all FTP and HTTP packets.

```
ADD -IP ClassifierList CL_ftp 20 FTP
ADD -IP ClassifierList CL_http 20 HTTP
```

- 2 Define the OUTbound policy on port 3, applying the rate limit action associated with the above defined classifierlist. All FTP and HTTP outbound packets on port 3 will be limited to 256 and 128 respectively. All packets beyond the specified rate will be discarded.

```
ADD !3 -IPQos POLicy POL1 20 CL_ftp OUTbound RateLimit 256 256 4000
ADD !3 -IPQos POLicy POL2 20 CL_http OUTbound RateLimit 128 128 4000
```

- 3 Enable the outbound Quality of Service Policy filter with the following command:

```
SETD !3 -IPQos CONTrol=OutFilter
```

Example 3 A VPN setup could consist of many virtual ports bound to a single path. This VPN setup would prevent aggressive slows from a single virtual port from hogging the entire bandwidth of a link. The virtual ports can be IPIP/PPTP/L2TP tunnels, and the parent port is FR/PPP.

The configuration steps below limit the rate of "outbound" traffic over virtual ports V1, V2, and V3. The virtual ports are IPIP tunnels bound to port 3. R1 is the bridge/router in which Quality of Service polices are applied.

To setup CBQ with rate limiting on the virtual ports, follow these steps:

- 1 First we need to setup the IP addresses on ports 1 and 3 for R2.

```
SETDefault !1 -IP NETaddress=20.1.1.1
SETDefault !3 -IP NETaddress=130.1.1.2
```

- 2 Create the IP termination addresses for the IPIP tunnel.

```
ADD !3 -IP NETaddress 130.3.1.2
ADD !3 -IP NETaddress 130.2.1.2
```

- 3 Create the IPIP virtual ports and IP addresses.

```
ADD !v1 -Port VP IPIP 130.1.1.1
ADD !v2 -Port VP IPIP 130.2.1.1
ADD !v3 -Port VP IPIP 130.3.1.1
SETDefault !v1 -IP NETaddress=150.1.1.2
SETDefault !v2 -IP NETaddress=150.2.1.2
SETDefault !v3 -IP NETaddress=150.3.1.2
```

- 4 Create the IP addresses for ports 1 and 3 for R1.

```
SETDefault !1 -IP NETaddress=10.1.1.1
SETDefault !3 -IP NETaddress=130.1.1.1
```

- 5 Create the IPIP Termination IP addresses.

```
ADD !3 -IP NETaddress 130.3.1.1
ADD !3 -IP NETaddress 130.2.1.1
```

- 6 Create the IPIP virtual ports and IP addresses.

```
ADD !v1 -Port VP ipip 130.1.1.2
ADD !v2 -Port VP ipip 130.2.1.2
ADD !v3 -Port VP ipip 130.3.1.2
SETDefault !v1 -IP NETaddress=150.1.1.1
SETDefault !v2 -IP NETaddress=150.2.1.1
SETDefault !v3 -IP NETaddress=150.3.1.1
```

- 7 Setup static routes so that traffic will go over each individual tunnels.

```
ADD -IP ROUTe 20.1.1.3 !v3 1
ADD -IP ROUTe 20.1.1.2 !v2 1
ADD -IP ROUTe 0.0.0.0 !v1 1
```

- 8 Create a classifier list which will match all packets for the default gateway.

```
ADD -IP ClassifierList CL_any 10 to 0.0.0.0/0
```

- 9 Define each virtual port to CBQ1 class queue.

```
ADD !v1-!v3 -IPQos POLicy POL1 10 CL_any OUTbound
ClassBasedQue CBQ1
```

- 10 Enable the outbound Quality of Service Policy filter with the following command:

```
SETDefault !v1-!v3 -IPQos CONTrol=OutFilter
```

- 11 Define the Class Based queue "CBQ1" for port 3. The bandwidth defined as 1 Mbps.

```
ADD !3 -Port ClassBasedQue ROOT NULL 1000
ADD !3 -Port ClassBasedQue CBQ1 ROOT 1000 def
```

- 12 Enable the queue control for port a, for CBQ.

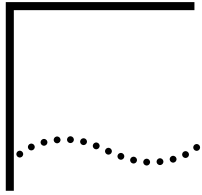
```
SETDefault !3 -Port QueueCONTrol=ClassBasedQue
```

- 13 Define the rate limiting parameters for each virtual port, associated with the class base queue CBQ1.

```
ADD !v1-!v3 -Port RateLimit CBQ1 64 128 2000 def
```

- 14 Enable the queue control for each virtual port as CBQRL.

```
SETDefault !v1-!v3 -Port QueueControl=CBQRateLimit
```



CONFIGURING SYSTEM IP

The System IP is a software-only interface that emulates an IP interface. It is always active (“up”) and allows IP operations on the NETBuilder bridge/router, as long as at least one interface on the bridge/router is active.



Only one System IP is allowed per NETBuilder bridge/router, and only IP-based applications are supported by this feature.

A System IP is similar to a regular IP interface except that it is not attached to any physical port, and it is always “up” as long as at least one interface on the NETBuilder bridge/router is active. By configuring a System IP in your Enterprise OS software, you do not need to know the specific address of the active interface to be able to Telnet to the NETBuilder bridge/router; you need only remember the System IP.

After a System IP is created, most IP-based operations using the System IP’s IP address can be performed. The following applications benefit from the use of a System IP:

- IP applications, such as Telnet, SNMP, and FTP
- Tunnel endpoints, such as X.25 over TCP, and GRE
- OSPF router ID

Configuring System IP

To configure the System IP, enter:

```
SETDefault -IP SystemIP = 1.1.1.1
```



For the System IP to be valid (“up”), at least one interface on the NETBuilder bridge/router must be active. The chosen IP address can be:

- *One of the active IP interface’s IP addresses.*
- *Part of an active IP interface’s subnet.*
- *Different from all active IP interfaces.*

When you create the System IP address, an entry is added to the route table. The route table must contain this entry so that you can issue a ping command from the NETBuilder bridge/router to the System IP. Table 19 lists the information contained in the entry added to the route table.

Table 19 Information Added to the Route Table by System IP

Gateway	Any active (“up”) IP interface
Metric	1 (the “owner” is System IP)

NB2*	<p>Added to the ARP table to allow the NETBuilder bridge/router to respond to ARP requests on the System IP.</p> <p>The MAC address in the ARP response is the active IP interface that has a network number that matches the System IP's network number.</p>
------	---

* This will be employed only if the System IP has the same network number as one of the active interfaces.

The following are examples of RIP and ARP table entries based on System IP address usage.

When the SystemIP is set to 4.4.4.4 and there is no routing entry for network 4.0.0.0, the following entry is automatically added to the routing table:

```
4.4.4.4 255.255.255.255 3.3.3.52 1 Up 0 SystemIP
```

When the SystemIP is set to 3.3.3.3 and there is an interface which already has 3.3.3.52 configured, the following new entry in the ARP table is made:

```
3.3.3.52 2F Local %08000205734A Ethernet Static
3.3.3.3 2F Local %08000205734A Ethernet SystemIP
```

To delete the System IP, enter:

```
SETDefault -IP SystemIP = 0.0.0.0
```

When you issue this command, the System IP's route table entry and ARP table entry are removed. The routing advertisements that System IP had sent out are flushed automatically by the neighboring routers.



The SETDefault -IP SystemIP command can also be used to change the IP address of an existing System IP.

Routing Issues

The Enterprise OS routing engine takes care of outgoing packets that have the System IP as their source IP address. For incoming packets that have the System IP as their destination IP address, the System IP's route needs to be advertised to neighboring networks.

To advertise to neighbors, routing packets need to be sent out periodically. There are several methods to accomplish this:

- If the System IP address is the same as the active IP interface or has the same subnet as the active IP interface:
 - Nothing needs to be done. For instance, if this interface is subsequently disabled, a host route advertisement for each of the IP interfaces with a gateway IP address the same as the IP interface's address does not need to be issued.

Table 20 System IP Address with the Same Subnet as an Active IP Interface's IP Address

	Port "UP" (Active)	Port "DOWN" (Inactive)
OSPF	<ol style="list-style-type: none"> 1 If OSPF is enabled: do nothing. 2 If OSPF is disabled: <ul style="list-style-type: none"> ■ If DIR NET, do nothing. ■ If not DIR NET, send Host Route advertisement to other interfaces. 	Send Host Route advertisement to other interfaces.
RIP/RIPv2	<ol style="list-style-type: none"> 1 RIP Enabled: do nothing. 2 RIP Disabled: Send subnet route advertisement to other interfaces. 	Send Host Route advertisement to other interfaces.

- If the System IP address is different than all active IP interfaces:
Send out a route stub (host) advertisement for all IP interfaces with the gateway IP address of the IP interface's address. Table 21 summarizes the necessary actions for all routing protocols supported.

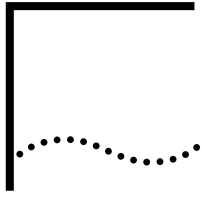
Table 21 System IP Different from All Active IP Interfaces IP Addresses

OSPF	Send Host route advertisement to all interfaces.
RIP/RIPv2	Send Host route advertisement to all interfaces.

ARP Packets ARP packets for the System IP's MAC address are generated only if the System IP address has the same subnet as one of the IP interfaces. In this case, the MAC address of that interface is used.

NAT/Firewall Issues Currently, all NAT/Firewall policies are on a per-port basis, which creates a lack of NAT/Firewall issues on the System IP. All NAT/Firewall policies are applied at the port level.

VRRP Issues The System IP address of a router should not be used as a VRRP Virtual Router IP (VIP). Doing so may cause two routers to respond to the same IP address. When sending a request (such as SNMP) to the VIP, the response can be returned from either of the routers.



CONFIGURING VIRTUAL PRIVATE NETWORKS

This chapter describes virtual private networking and how to use a PathBuilder switch with Enterprise OS software to configure a virtual private network (VPN).

Remote Access Alternatives

VPNs are a cost-effective alternative for providing remote access or remote office connectivity to a central site.

Typically a company is required to use dedicated leased lines, packet-switching services, and/or direct dialup connections to enable remote users and remote offices to connect to a central site. A VPN provides a less expensive method of providing this connectivity.

The internet service provider (ISP) is an important element in of a VPN. By providing local access for any remote user or remote office, the ISPs network replaces the leased lines, packet-switching services, and direct dialup connections. Instead of directly managing remote access WAN lines a company can outsource this responsibility to an ISP, resulting in fewer WAN issues to track and potentially significant cost savings.

Using Tunnels

To ensure security and multiprotocol support a tunnel is created to the central site. Tunneling allows you to encapsulate IP and non-IP packets, to provide security using IPsec, and to obtain access to the central site network through a firewall.

A tunnel can be set up in one of two ways:

- From the ISP to the central site. This configuration is used to connect individual remote users to a central site.
- From a remote site to the central site. This configuration is used to connect a remote office to a central site.

ISP to Central Site Tunneling

The ISP must have tunnel-enabled access servers, like the Total Control hub, if the remote clients cannot support the tunneling protocol.

In this configuration, the tunnel set up proceeds as follows:

- First the remote user dials into the ISP's access server.
- The access server recognizes (based on a user ID, for instance, or on the user's choice from a menu) that this connection should be tunneled to the central site.
- The access server establishes the tunnel with the central site.

- The remote user then establishes a session directly with the central site via the tunnel, just as if the two were directly attached.

While this configuration has the advantage that no special software is required on the remote user, the remote user can dial only into properly equipped access servers.

Remote User to Central Site Tunneling

In this configuration, the remote user (the client), such as an OfficeConnect NETBuilder bridge/router or an appropriately configured personal computer, supports the tunneling protocol. The ISP does not have to support tunneling in any way.

The remote user dials the ISP, but once the connection is set up, the remote user and the central site establish the tunnel, using authentication based on a user ID and password and perhaps on a digital certificate.

The remote user and the central site may also negotiate encryption. After the tunnel has been established, communications proceed as if the ISP were not mediating the connection.

Creating a VPN for Individual Remote Users

In place of setting up multiple remote access servers at the central site, VPNs allow remote users to dial a local ISP. Using a VPN for remote access is particularly useful if you have remote users at a great distance from the central site. For example, users in Europe can call a local number instead of dialing in to the central site in New York.

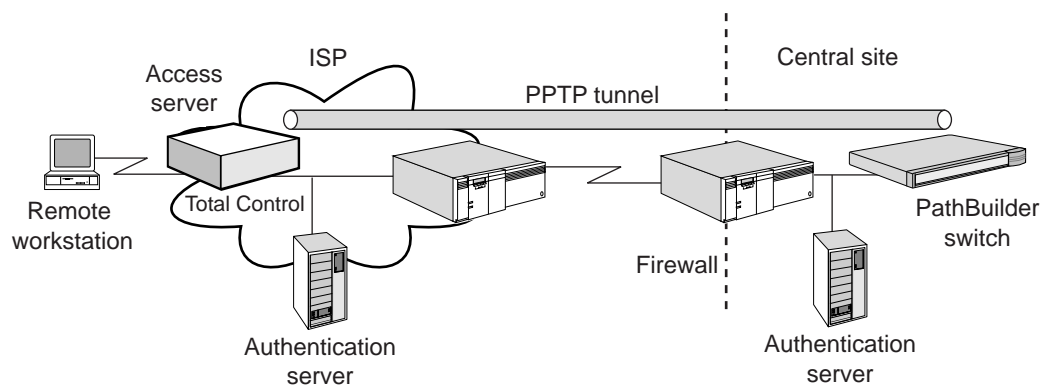
The following two examples show remote access VPN configurations.

- Example 1** In Figure 81, the ISP is configured to create a tunnel from the ISP's access server to the central site.



This method can also be used for a remote office if you do not want to configure tunneling on the bridge/router at the remote office.

Figure 81 ISP to Central Site Tunnel



The connection process typically follows this order:

- The remote user (the client) dials the ISP.
- The ISP assigns an IP address to the remote user client.

- The ISP checks its authentication server for the user, and creates a PPTP (or L2TP) tunnel to the central site based on authentication data.
- The central site checks its authentication server to verify that this user can access the network and forwards the data.



No special configuration is required on the remote user computer except the configuration required to dial into the ISP's access server.

At the central site, follow these steps:

- 1 Configure the L2Tunnel Service (see the Configuring L2Tunnel Connections chapter in *Using Enterprise OS Software*) to enable the PathBuilder switch as a tunnel terminator.
- 2 Configure the firewall device (if present) or the PathBuilder switch to allow tunnel traffic through (see the Building Internet Firewalls chapter in *Using Enterprise OS Software*).
- 3 Configure the RAS service to allow authentication of the user by a server, such as a RADIUS server (see the Configuring Remote Access Services chapter in *Using Enterprise OS Software*).

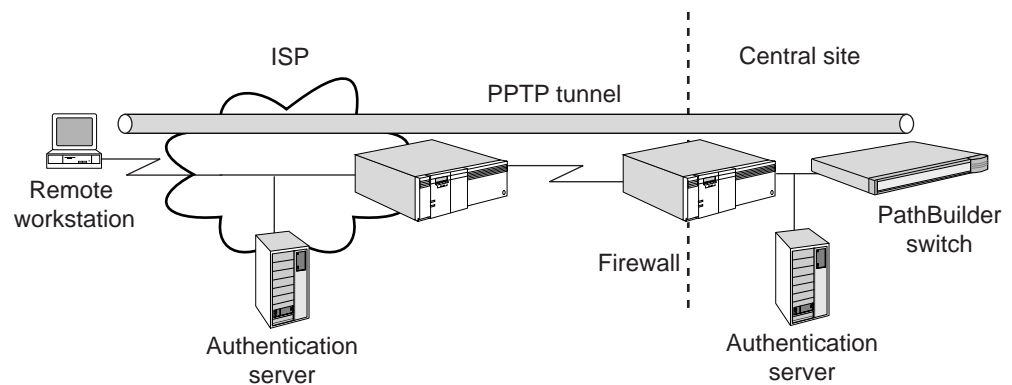


The firewall and RAS functions can also be configured on the PathBuilder switch. The configuration example in Figure 81 shows these services being performed on separate devices, for purposes of clarity.

Example 2 In Figure 82, the remote workstation is configured to create a tunnel directly to the central site.

See the documentation for your workstation or consult your operating system vendor for instructions on how to configure your workstation as the remote PPTP/L2TP client.

Figure 82 Remote Workstation to Central Site Tunnel



The connection process typically follows this order:

- The remote client dials the ISP.
- The ISP assigns an IP address to the client.
- The remote client sends data to the IP address of the central site.

- The Windows 95/NT workstation client creates a PPTP tunnel to the central site based on authentication data.
- The central site checks its authentication server to verify that this user can access the network and forwards the data.

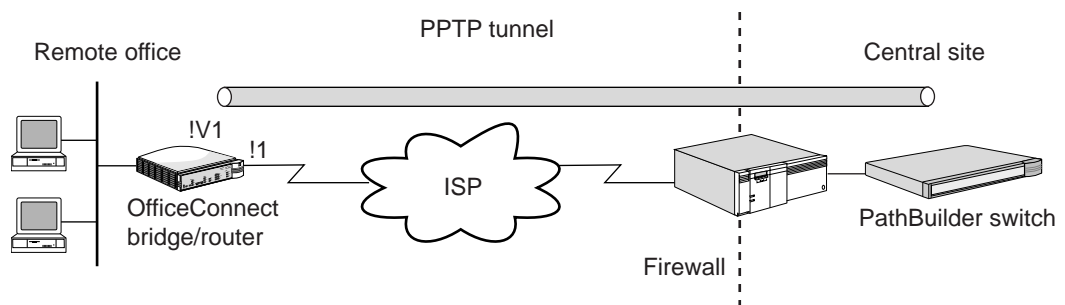
At the central site, follow these steps:

- 1 Configure the L2Tunnel service (see the Configuring L2Tunnel Connections chapter in *Using Enterprise OS Software*) to enable the PathBuilder switch as a tunnel terminator.
- 2 Configure the firewall device if present, or the PathBuilder switch, to allow tunnel traffic through (see the Building Internet Firewalls chapter in *Using Enterprise OS Software*).
- 3 Configure the RAS service to allow authentication of the user by a server, such as a RADIUS server (see the Configuring Remote Access Services chapter in *Using Enterprise OS Software*).
- 4 Enable PPP encryption to allow encryption keys to be used by MPPE (see the Configuring Wide Area Networking Using PPP chapter in *Using Enterprise OS Software*).

Creating a VPN for a Remote Office

You can create a VPN to connect a remote office PathBuilder switch to the central site through the ISP using tunneling protocols such as the point-to-point tunneling protocol (PPTP). Figure 83 shows a typical configuration. In this configuration, the tunnel is established between the remote office and the central site. The ISP provides access to the shared network but does not interact in the tunneling setup.

Figure 83 Remote Office Tunnel



The connection process typically follows this order:

- The remote office OfficeConnect NETBuilder bridge/router dials the ISP.
- The ISP assigns an IP address to the remote office bridge/router.
- The remote office OfficeConnect NETBuilder bridge/router sends data to the IP address of the central site.
- The data is encrypted using IPsec.
- A PPTP/L2TP tunnel is created between the remote site and the central site, and the data is forwarded through the firewall of the central site.
- The data is decrypted by the central site.

**On the Remote Office
OfficeConnect
Bridge/Router**

On the OfficeConnect NETBuilder bridge/router, follow these steps:

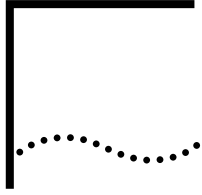
- 1 Configure dial-up to the ISP (see the Configuring Port Bandwidth Management chapter in *Using Enterprise OS Software*) or virtual leased line configuration (see the Configuring L2Tunnel Connections chapter in *Using Enterprise OS Software*).
The ISP assigns an IP address to the client, or you configure an IP address that is applicable to the ISP's network using the IP service (see the Configuring IP Routing chapter and the Configuring Network Address Translation chapter in *Using Enterprise OS Software*).
- 2 Create a virtual port specifying the SysCallerID (SCID) of the central site PathBuilder switch (see the Configuring Port Bandwidth Management chapter in *Using Enterprise OS Software*).
- 3 Add a dial number list to the virtual port specifying the IP Address of the central site and the type PPTP (see the Configuring Port Bandwidth Management chapter in *Using Enterprise OS Software*).
- 4 Configure the L2Tunnel Service (see the Configuring L2Tunnel Connections chapter in *Using Enterprise OS Software*) to enable the bridge/router as a tunnel initiator.
- 5 Configure IPsec on the virtual port specifying the same profile contents and key at the central site (see the Configuring IPsec chapter in *Using Enterprise OS Software*).

After the remote site dials the ISP, any data that is sent to the IP address of the central site creates a PPTP tunnel between the two sites.

**On the Central Site
PathBuilder Switch**

At the central site, follow these steps:

- 1 Configure the L2Tunnel Service (see the Configuring L2Tunnel Connections chapter in *Using Enterprise OS Software*) to enable the PathBuilder switch as a tunnel terminator.
- 2 Configure the firewall device if present, or the PathBuilder switch to allow tunnel traffic through (see the Building Internet Firewalls chapter in *Using Enterprise OS Software*).
- 3 Configure IPsec specifying the same profile contents and key as the remote site (see the Configuring IPsec chapter in *Using Enterprise OS Software*).



CONFIGURING PUBLIC KEY INFRASTRUCTURE

This chapter presents an overview of Public Key Infrastructure (PKI) and describes how to configure PKI on your device.

Applications including Internet Key Exchange (IKE) and Secure Sockets Layer (SSL) use public key technology for security purposes. These purposes include identifying oneself to remote entities, verifying a remote entity's identity, and initiating secure communications with remote peers. Such applications require a public key infrastructure (PKI) to securely manage public keys for widely-distributed users or systems. The X.509 standard, implemented by the 3Com Enterprise Operating System, is a widely accepted basis for a PKI infrastructure. X.509 defines data formats and procedures related to the distribution of public keys using certificates digitally signed by Certificate Authorities (CAs).

Overview of PKI

PKI addresses the management of public and private key-pairs and the certificates associated with those key-pairs. Certificates are signed electronic documents that certify that a specified public key belongs to the entity named on the certificate.

For a 3Com Enterprise OS device, the entity is the device itself, and the name on the certificate is the device's IP address, DNS name, or a combination of both. A trusted third party called a Certificate Authority (CA) signs the certificate, certifying that the named entity possesses the key-pair. The format of certificate is defined in the X.509 standard.

PKI supports the following functions:

- Key-pair management
- Generation of certificate requests (which can then be delivered to the CA)
- Addition and deletion of certificates to and from a database
- Validation of the trustworthiness of a certificate chain (including signature validation) to verify the authenticity of a remote entity's public key
- Certificate revocation

Key-Pair Management

For secure communication using public key technology, each participating device needs a key-pair. The key-pair consists of two components; the private component and the public component. The private component of the key-pair should be *secured*, because the security offered by public key technology depends on the confidentiality and integrity of the private component of the key-pair.

The public component of a device's key-pair is advertised to any other entity that tries to engage in secure communication with that device. Using PKI, this advertisement is in the form of a X.509 certificate that contains the device's public key.

The device uses RSA public and private key-pairs. These are created on the device using the RSA-provided BSAFE library. The key-pair is stored on the local flash or disk. The private component of the key-pair is further secured by saving only the encrypted version, using a secret key approach called the key-encryption-key (KEK). The private component of the key-pair exists unencrypted only when it is in active use. For more information on KEK, see the Commands chapter in the *Reference for Enterprise OS Software*.

Certificate Enrollment Certificate enrollment results in the end device acquiring a valid certificate of its own. The major steps in certificate enrollment are:

- 1 The device acquires the root CAs certificate out-of-band, and installs it into the local certificate database (onto the device's flash or disk).

This step assumes that you have verified the integrity of the CA's certificate. The trustworthiness of a certificate chain to be validated depends on the trustworthiness of the root CA's certificate (which is normally self-signed).

- 2 The device generates its own public/private key-pair and installs the private-key component into a secure local database.
- 3 The device issues a request for its own certificate to the CA.

The request is for a certificate that contains the device's own public key to be authenticated by the CA. The certificate request can be transported to the CA either out-of-band or using a transport mechanism, such as HTTP or FTP.

- 4 The CA checks its policy to see if the requesting device is authenticated. If so, the CA issues a certificate to the device and optionally adds it to a publicly accessible certificate repository (for example, an LDAP directory).
- 5 The device fetches the certificate.

Certificate Installation and Storage After the certificate is downloaded to the device, it needs to be installed. Installation indicates that the certificate is stored within the file system of the device in a place where it is used for public key transactions by the device's applications.

The following different kinds of X.509 certificates need to be stored in the file system on the device:

- The device's own certificates, which are issued by a CA.
- Certificates of trusted CAs. These are usually fetched as a file that contains the CA's certificate.
- If required, a remote peer's certificate may also be stored locally (such as an untrusted certificate that must be validated as part of a certificate chain terminating on a trusted CA certificate).

Certificate Usage and Validation Certificates are used to securely bind a device's identity to its public key. A remote entity (the verifying party) that wishes to securely interact with a device, for example, device A, does the following:

- Fetch device A's certificate.
- Verify its authenticity using certificate chain validation.
- Extract the public key from device A's certificate.

- Start using device A's public key (for example, to verify a signature).

Device A provides either its certificate or a chain of certificates to the verifying party. A chain of certificates starts with the device's own certificate, which then points to the certificate of the CA that signed the device's certificate. This CA certificate can point to another certificate of another CA that signed the first CA's certificate, and so on. The chain can continue until a certificate is reached that belongs to a CA that is considered trusted by the verifying party.

Normally, the certificate chain provided by Router A to the verifying party does not contain the trusted CA certificate. The trusted CA certificate must be available to the verifying party through an out-of-band mechanism, as described under step 1 of "Certificate Enrollment." The chain is minimally validated by the following actions:

- Validating that all certificates are well-formed (a supported version number and signature algorithm)
- Verifying the validity times on each certificate in the chain
- Verifying the signatures on all subordinate certificates of the chain.

Further checks involve the validation that no certificate in the chain has been revoked. Additionally, other attributes relating to the policy of certificate use can be validated for all certificates in the chain.

Certificate Revocation

Each certificate has its own validity period during which the information contained in the certificate is guaranteed by the certificate issuer (CA) to be accurate. Certificate revocation is the process of informing users that the information contained in a certificate has become unexpectedly invalid (for example, by the loss of the private key or by private key compromise).

The most common methods used to revoke certificates is to issue a Certificate Revocation List (CRL). A CRL is a list of revoked certificates that is periodically issued by the CA. To be assured of the validity of a certificate, you must check the latest CRL to see if the certificate has been revoked.

Preparation for a Public Key Infrastructure

Within an enterprise, a PKI can support the key and certificate management of a large number of devices that communicate securely among themselves. Some of the devices are devices. This chapter focuses only on the configuration of PKI for devices. However, various device combinations using PKI for secure interaction within the enterprise are possible:

- Router-to-router interaction using IPSEC VPN tunnels.
- Workstation client-to-router interactions for secure tunnelled VPN RAS connections.
- Workstation client-to-server interactions for secure tunnelled VPN connections.
- Workstation client-to-server interactions for secure SSL/TLS web connections.

The integration of a set of devices into an enterprise PKI environment requires proper planning that goes beyond the scope of this chapter. Minimally, proper planning requires answers to the following questions:

- What types of applications and devices are being secured by the PKI? Is the PKI required to secure IPSEC/IKE interactions? Are there SSL/TLS interactions?
- What is the scope of the devices being secured by the PKI? Is this only an intranet? Will there be extranets with other enterprises having possibly different PKI organizations and certificate authorities?
- What CA configurations will be provided to issue the certificates that will be loaded onto the devices within the enterprise? Who will be the CA vendor? Will the CA service be provided in-house? Outsourced? A combination of in-house and outsourced CA function? Will the CA service be distributed within the enterprise using multiple domains of trust, or will there be a single, centralized CA using a single domain of trust within the enterprise?

PKI Applications and Devices

You need to determine the types of devices and the applications running on those devices that will use public key technology for security. You must also select a PKI to manage that technology. Consider the following questions:

- Will the devices be routers, client workstations, or servers?
- What security applications are being supported (for example, IPsec/IKE, SSL/TLS)?
- What combinations of devices are being supported?

The types of certificates that must be issued to the various devices are determined by the types of supported applications.

In addition to public key, device name, validity date, and signature information, a certificate can carry various types of policy information. Policy information includes the types of applications for which a certificate is to be used. To determine the types of information that will be entered into certificates you should understand the types of applications supported, and the security policy (called the *trust policy*) employed in using the certificate.

Scope of the PKI

You must also determine the security perimeters for which public key-based security is to be implemented. Consider the following questions:

- Will the public key security domain encompass only the enterprise (for example, an intranet) including the logical network of any enterprise employees accessing the enterprise network through firewalls?
- Will interactions using public key technology occur across enterprise boundaries between enterprises (for example, extranets)?

If only a single enterprise is involved, certificates used by the devices are issued only by CAs directly under the control of that enterprise. If extranets are involved, mechanisms must be in place to establish trust for certificates issued by CAs under the control of the external enterprises. This is also known as cross-certification.

Certificate Authority Configuration

The CA issues the certificates used by the devices being managed in a PKI. Several decisions regarding how the CA function will be implemented must be made:

- Will there be a single CA for the enterprise, or several CAs?

In the former case, there is a single domain of trust in which all non-expired, non-revoked certificates can be trusted by all devices; because all devices trust the signature of the single issuing CA. In the latter case, there are multiple

domains of trust either within the single enterprise or among several external enterprises (for extranet configurations).

- If you are using multiple CAs within a single enterprise, how should the enterprise trust domains be divided up (for example, by corporate divisions, by corporate functions, and so on)? Should the subsidiary CAs be established in a hierarchy under the control of a central "root" CA, or should the enterprise CAs be equal peers with trust agreements between them (using cross-certification)?
- Who will provide the CA service?

This can be done either completely in-house, outsourced, or via a hybrid of in-house and outsourced facilities. In an in-house configuration, the enterprise builds the CA either on its own or using CA server products offered by various CA vendors. The management of the CA is performed completely by the enterprise.

In an outsourced configuration, a CA service organization (such as Verisign) contracts with you to provide the CA services. In a typical hybrid configuration, the enterprise provides the service of authenticating the users and devices that will be issued certificates (called the Registration Authority function or RA) and passes on only authenticated certificate requests to a CA outsourcing service, which generates and signs the certificates to be issued.

- What vendor should provide the CA facilities?

These facilities come in many forms, from toolkits that allow the enterprise to build a CA server to issue certificates, to packaged CA servers that just need to be configured for the enterprise's environment, to outsourced CA services. There are many CA vendors of toolkits, packaged server products, and outsourced CA services.

3Com supports CA products and services from the following vendors:

- Entrust – Provides a packaged CA server product called "Entrust PKI Authority/Admin/Directory," along with a supplementary product called "VPN Connector," that allows VPN devices such as devices to enroll for certificates with the Entrust Authority CA.
- Verisign – Provides an outsourced CA service called "Onsite." The Verisign CA can be used only with the 3Com PKI Manager application. It cannot be used if you are configuring your device using the command line interface or the PKI dialog facility.

For more information on configuring the Entrust and Verisign CA products, see the *Enterprise OS Software Release Notes*.

Device PKI Configuration

After the major PKI planning decisions have been made and the CA facilities have been configured, you can enable PKI on the devices by generating the key-pair and installing the certificates.

You can select one of three mechanisms to configure a device for PKI operation:

- The PKI Manager application
- The device PKI dialog facility

- The device command-line interface

PKI Manager

The PKI Manager is a separately purchased product that is bundled with the Secure VPN Manager product from 3Com. The PKI Manager installation is described in the CD-ROM booklet that ships with the product. Details of the PKI Manager application are documented in the PKI Manager online help facility.

The PKI Manager largely automates the device PKI configuration using a graphical user interface. The PKI Manager is required if you wish to enroll a device with the Verisign Onsite CA service to generate the device's certificate. The PKI Manager can also be used if the Entrust Authority CA/VPN Connector product is used for the CA. In this case, the PKI Manager must be installed on the same system as the VPN Connector product.

Device PKI Dialog Facility

If the PKI Manager is not available, the device provides a dialog facility to configure PKI, including the generation of a key-pair and the enrollment of the device with a CA to secure its certificate. For enrollment, this dialog can only be used with the Entrust CA. It cannot be used for a Verisign CA.

The dialog facility can be invoked at the device console or using a remote Telnet session to the device. The facility simplifies the process of generating key-pairs, making certificate requests, and fetching and installing certificates.

Using the dialog facility, the key generation and enrollment procedure has three main parts:

- Use a Telnet session to generate a key-pair, generate a certificate request, and download the certificate request to the VPN Connector workstation.
- Use the VPN Connector application to request the Entrust Authority CA to generate a certificate from the certificate request that was downloaded from the device. Enrollment of the device using VPN Connector follows the procedures for PKCS10 certificate requests described in the Entrust documentation.
- Use a Telnet session to download the device's own certificate and the CA's certificate to the device, and install these certificates into the device's local certificate database.

Initiating the Dialog Facility

To start the PKI dialog facility enter the following command from the console prompt:

```
PkiCONFigure
```

A main menu is launched that contains a set of configuration options described in the following sections.

PKI Dialog Facility Main Menu Options

The PKI dialog facility main menu provides the following options:

- **Enrollment Key Management** – Used to specify an enrollment key. The enrollment key is used for automated authentication of the device to the PKI Manager and automated authentication of a trusted CA certificate to be

installed. The Enrollment Key Management option is only used if you are using the PKI Manager application.

- **Key-Pair Management** – Used to generate a new public/private key-pair, to delete the key-pair, or to display the public component of the key-pair.
- **Certificate Requests** – Used to generate a certificate request, delete a certificate request, or display the contents of the current certificate request.
- **Certificate Fetch** – Used to fetch a certificate to store locally or to install.
- **Certificate Install** – Used to fetch a certificate and install it into the local certificate database.
- **Certificate Display** – Used to display remote or locally installed certificates.
- **CRL Display** – Used to display the local CRL cache.
- **PKI File Display** – Used to display a file containing a PKI object (certificate, CRL, or public key) located on a remote repository or on the local file system.

Enrollment Key Management

The Enrollment Key (EK) is a shared secret key between the device and the PKI Manager. It is used to automate the process of device authentication to the PKI Manager application and automate the authentication of trusted CA certificates being installed in the device. The EK can only be entered using the PKI dialog facility; the command-line interface cannot be used.

The EK should originate with the central PKI administrator operating the PKI Manager. The EK application is a shared key between the PKI administrator entering the key locally at the PKI Manager and the device administrator entering the key locally at the device. Therefore the EK *must not* be transferred across an unsecured network.

Key-Pair Management

The key-pair management submenu has the following options:

- Generate a key pair
- Delete a key pair
- Display the public key

Certificate Requests

The certificate request submenu has the following options:

- Generate a certificate request
- Transfer a certificate request to a CA
- Delete a certificate request
- Display a certificate request

Certificate Fetch and Install

Certificate fetch and install are similar dialog options. Install prompts you to wait for the certificate to be generated. Fetch attempts to fetch the certificate immediately. You are prompted for the certificate profile to be used (a local name for the certificate), the type of certificate, and the URL used to fetch the certificate.

Certificate Display

The certificate display submenu has the following options:

- List profile names of installed certificates
- Display installed certificates in short form
- Display installed certificates in long form

The list option displays a certificate database directory listing. The long form displays all fields of the certificate, including the full DER encoding of the certificate. Both the long and short form displays show the MD5 and SHA1 fingerprints of the certificate(s) being displayed.

CRL Cache Display

The CRL cache display option displays the contents of the local internal CRL cache. The local internal CRL cache contains the contents of all current CRLs that have been used for certificate validation. The display shows the validity times for the CRL, the CRL issuer name, and a list of serial numbers of all certificates on the CRL.

PKI File Display

The PKI file display option displays a remote or local file to see if it is a supported PKI object. Supported PKI objects are certificates, CRLs, or public keys.

PKI Command Line Interface

The command line parameters for the PKI Service in the device are described in detail in the the PKI Parameters chapter in *Reference for Enterprise OS Software*.

The command-line interface provides the most flexibility and provides some extended functions that are not available using the PKI Manager or the PKI dialog facility. The PKI command-line interface allows you to perform the following tasks:

- Lock/unlock the PKI databases.
- Generate or delete RSA key-pairs for the device.
- Generate a PKCS-10 certificate request based on the device's identity and the public key of the RSA key-pair.
- Install and delete a certificate.
- Display PKI objects including certificates or CRLs, and the current PKI configuration.
- Configure CRL distribution points.
- Configure remote repository default addresses.
- Configure the trust policy used to validate certificate chains.

For ease of use, you should use the PKI Manager or the PKI dialog facility for basic key generation and certificate enrollment, rather than the PKI command-line interface.

PKI Database Lock/Unlock

To prevent accidental deletion of a key-pair or certificates, the PKI Service operates in a PKI database lock and unlock mode. To modify the key-pair or certificate information (such as deleting or regenerating the keypair, or deleting a certificate)

you must unlock the database. Normally, the database is locked to prevent the changes.

A certificate is initially installed into the local database when the database is unlocked. In this mode, the certificate is disabled. This means that the certificate is not yet available to be used by the security applications on the device. To enable the certificate for security applications, the database must be locked. Thereafter, the certificate remains enabled, independent of the state of the database.

To lock the database, enter:

```
SETDefault -PKI CONTROL = Locked
```

To unlock the database, enter:

```
SETDefault -PKI CONTROL = Unlocked
```

Key-pair Generation and Certificate Enrollment Commands

Command-line commands are available to perform the following actions:

- To generate the key-pair.
- To delete the key-pair.
- To generate a certificate request from the device's identity and public key.
- To transfer the certificate request.
- To fetch the device's or a trusted CA's certificate.
- To install the certificate(s) into the local certificate database (at which point they become usable to the device's public key security applications).

This section describes the sequence of tasks use to configure PKI enrollment using the command line interface.

Unlock the PKI Database

To modify the PKI databases in the device, the PKI database must first be unlocked using:

```
SETDefault -PKI CONTROL = Unlocked
```

Download and Install the CA Certificate Into Router

Transfer the CA's certificate to the device and install it into the device's trusted CA database using:

```
ADD -PKI CERTIFICATE <cert-profile> IssuerCA InputFile <remote-url>
```

where:

- **cert-profile** is the name assigned to the CA certificate.
- **remote-url** is a URL the device uses to fetch the CA certificate on the CA's filestore or on some other repository (for example, an LDAP directory).

Generate an RSA Key-Pair

Determine the appropriate key size for the device and generate an RSA key-pair using:

```
ADD -PKI KeyPair DEFAULT <key-size>
```

where:

key-size is the size of the key, in bits.

If this is a reenrollment process, rather than the first enrollment, first delete the existing key using:

```
DEL -PKI KeyPair DEFault
```

Set Up Parameters for the Certificate Request

Decide what parameters are to go into the certificate to be issued for this device. Choices to be made include:

- Will you use a domain name to identify the device within the certificate and, if so, what will it be?
- Will you use an IP address to identify the device within the certificate and, if so, what will it be?
- What encoding will you use for the certificate request?

If a domain name will be used to identify the device within the certificate, use:

```
SETDefault -PKI DNS=<router-domain-name>
```

where:

router-domain-name is the selected domain name for the device.

If an IP address will be used to identify the device within the certificate, use:

```
SETDefault -PKI IPADDRESS = <ip-address>
```

where:

- **ip-address** is the selected IP address for the device

To specify the method by which the certificate request is to be encoded, use:

```
SETDefault -PKI ENCODING = DER | BASE64 | PEM
```

Generate and Transfer the Request for the Router's Certificate

After the certificate request parameter information has been configured, the device must generate the certificate request and transfer it to a specific location within the CA filestore using:

```
ADD -PKI CertReq CertReqFile <remote-req-url> CertReqFP  
<remote-fingerprint-url>
```

where:

- **remote-req-url** is a URL that specifies the desired method of transfer, and the location of the filestore, and the location within the filestore where the certificate request is to be transferred.
- **remote-fingerprint-url** is a URL that specifies the desired method of transfer, the location of the filestore, and the location within the filestore

where the secure certificate request fingerprint is to be transferred. You can omit this parameter if the secure certificate request fingerprint is not used.

Acquire the Router's Certificate from the CA/RA

Next, complete the backend procedures with the CA/RA to have the device's certificate generated. This task varies, depending on the type of CA being used for enrollment (Entrust or Verisign).

Add the Router's Certificate to the Local Certificate Database

When the device's certificate becomes available from the CA, download certificate to the device, and install it in the device's local certificate database using:

```
ADD -PKI CERTificate <cert-profile> Self InputFile <remote-url>
```

where:

- **cert-profile** is the name to assign to the device's certificate.
- **Self** indicates that the type of certificate being installed is the device's own certificate.
- **remote-url** is a URL where the device will retrieve the certificate. For example, the local filestore, an LDAP directory, or even a filestore assigned by the CA.

Relock the PKI Database

To finish the procedure, relock the database using:

```
SETDefault -PKI CONTrol = Locked
```

The device PKI is now available to service certificate enabled security applications running on the device (for example, IPSec/IKE).

Certificate/CRL Display	You can display the contents of either installed certificates or unknown certificates. The unknown certificate display is used to determine whether you want to install a particular certificate. You can also display a remote CRL file or the contents of the CRL cache.
CRL Distribution Points	The method to fetch CRLs and their location on remote repositories are specified by CRL distribution points. These can be obtained dynamically within certificates by way of the CRL distribution point extension (if present in the certificate) or statically by command-line configuration.
Remote Repository Default Addresses	Because the device must fetch certificates and CRLs from remote repositories, you must specify the method of fetch and the address of the repository. The commands use URLs to specify the method of fetch and location of certificates or CRLs. The addresses of the repositories where these are stored can be specified using an IP address or a DNS name. These addresses can be set to a default. Default addresses can be set for FTP, HTTP, or LDAP directory repositories independently.
Trust Policy Configuration	Remote devices participating in public key security applications with the local device must authenticate themselves to the local device. To do so, the remote devices normally pass their public key certificate to the local device. The validity of

the certificate depends on successful completion of a certificate trust validation check by the local device. The trust validation procedure uses certain fixed methods (such as certificate date validity checks and signature checks) and other configurable trust manager policy checks. You can specify the configurable trust management policy using the command-line interface.

PKI Configuration Example

To configure PKI on a central site device, follow these steps:

- 1 Unlock the database by entering:

```
SETDefault -PKI CONTROL=Unlocked
```

- 2 Create a PKI key-pair by entering:

```
ADD -PKI KeyPair 1024
```

```
Generating a <1024>-bit RSA key-pair. This may take some time.....
System's RSA key-pair generated.
```

```
Saving RSA key-pair ....
```

```
Encoded Public Key Hash:
```

```
0B:DD:17:DA:23:83:3F:6F:A0:D1:29:3A:64:C3:60:9F
```

- 3 Configure the IP address of the FTP Server by entering:

```
SETDefault -PKI FTPServer=129.213.40.190
```

- 4 Configure the DNS name by entering:

```
SETDefault -PKI DNSName=domain.name
```

- 5 Create a PKI certificate request by entering:

```
ADD -PKI CertReq toes CertReqFile ftp:///toes.req
```

```
Generating Certificate Request ...
```

```
Successfully generated certificate request
```

```
No Enrollment Key: No certificate request secure fingerprint generated.
```

```
Transferring Certificate Request via FTP to:
```

```
Server: 129.213.40.190
```

```
Pathname/Filename: /toes.req
```

```
254 bytes transferred. File transfer complete.
```

```
Certificate Request (DER) MD5 Fingerprint:
```

```
EF:AA:B1:DA:D0:CA:5C:56:88:B0:92:6D:AA:E9:F6:CB
```

```
Certificate Request (DER) SHA1 Fingerprint:
```

```
99:50:BD:93:87:4B:9C:9E:94:20:B5:A9:3A:4E:98:B5
```

```
2C:26:FD:7E
```

```
Verify these fingerprints with the CA.
```

- 6 Carry out the CA-specific method to generate the certificate from the certificate request. In this example, it is assumed that the CA puts the new certificate onto the FTP server as "toes.out".

- 7 Install the PKI certificate on the device by entering:

```
ADD -PKI CERTIFICATE toes.cer Self InputFile ftp:///toes.out
```



```
Fetching Certificate via FTP from:
  Server:          129.213.40.190
  Pathname/Filename: /toes.out
```

```
643 bytes transferred.  File transfer complete.
Certificate installed in local database.
```

You must lock the PKI database via the PKI CONTROL parameter for installed certificate to be usable.

8 Lock the PKI database by entering

```
SETDefault -PKI CONTROL=Locked
```

9 Add the device's CA PKI certificate by entering:

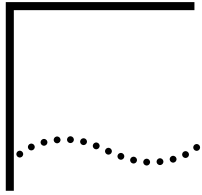
```
ADD -PKI CERTificate cacert IssuerCA InputFile ftp:///vpnconcacert.bin
Fetching Certificate via FTP from:
  Server:          129.213.40.190
  Pathname/Filename: /vpnconcacert.bin
```

```
685 bytes transferred.  File transfer complete.
```

```
This is a cert installed as a trusted root CA certificate.
WARNING: You should check cert fingerprints with CA.
         If the fingerprints do not match, then you must delete
         this certificate from the local database before
         reenabling the PKI service.
```

```
Certificate installed in local database.
```

You must lock the PKI database via the PKI CONTROL parameter for installed certificate to be usable.



CONFIGURING PROTOCOL INDEPENDENT MULTICAST-SPARSE MODE

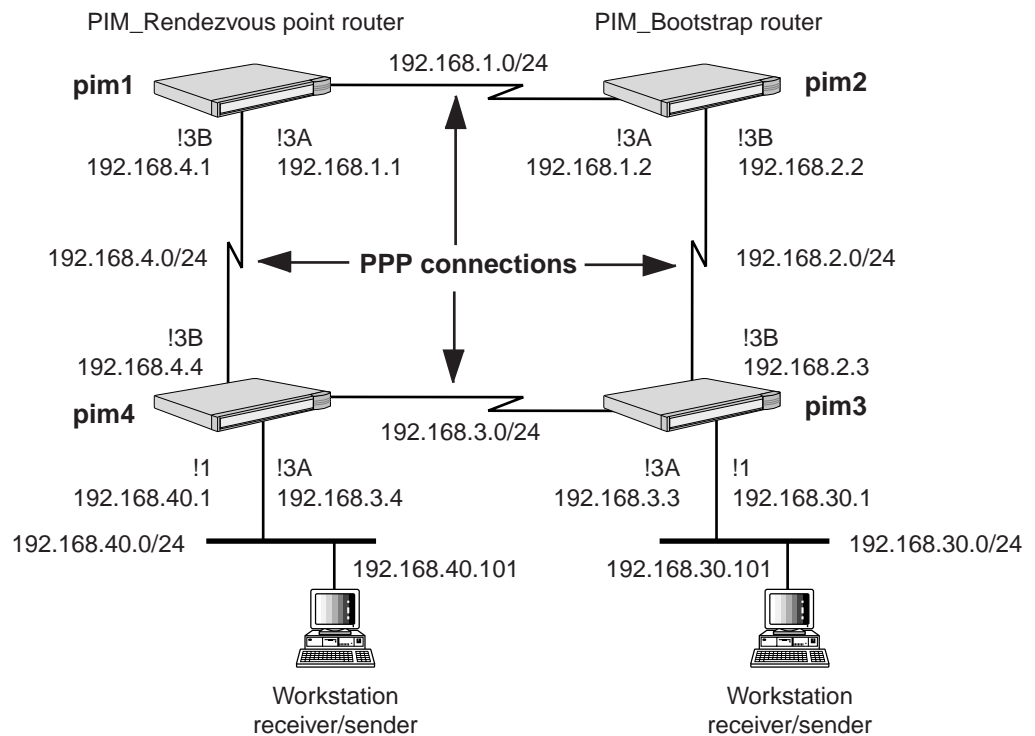
This chapter describes how to configure Protocol Independent Multicast-Sparse Mode (PIM-SM). This chapter provides a conceptual overview of PIM-SM and gives guidelines for operating and managing PIM-SM successfully.

For conceptual information, see “How PIM-SM Works” later in this chapter.

Configuring PIM-SM

This section contains an sample configuration and procedures for configuring that example. Figure 84 shows an sample PIM-SM configuration.

Figure 84 PIM-SM Sample Configuration



Configuring PIM 1 To configure the bridge/router labeled PIM1, follow these steps:

- 1 To set the baud rate and enable the paths enter:

```
SETDefault !3a,!3b -PATH BAud=256 CONTrol=e
```
- 2 To configure PPP links enter:

```
SETDefault !3a,!3b -PORT OWNer=ppp
```
- 3 Enable IP routing by entering:

- ```
SETDefault -IP CONTROL=ro
```
- 4 Configure the console prompt by entering:
 

```
SETDefault -SYS NMPrompt="pim1 #"
```
  - 5 Configure IP addresses by entering:
 

```
SETDefault !3a -IP NETaddr=192.168.1.1
SETDefault !3b -IP NETaddr=192.168.4.1
```
  - 6 Enable MIP control by entering:
 

```
SETDefault -MIP CONTROL=e
```
  - 7 Enable OSPF by entering:
 

```
SETDefault !3a,!3b -OSPF CONTROL=e
```
  - 8 Enable PIM-SMv2 by entering:
 

```
SETDefault !3a,!3b -PIM CONTROL=sm
```
  - 9 Enable CandidateRP by entering:
 

```
SETDefault -PIM SMControl=crp
```
  - 10 Add a CandidateRP address by entering:
 

```
ADD -PIM CandidateRP 192.168.1.1
```

**Configuring PIM2** To configure the bridge/router labeled PIM2, follow these steps:

- 1 Set the baud rate and enable the paths by entering:
 

```
SETDefault !3a,!3b -PATH BAud=256 CONTROL=e
```
- 2 Configure PPP links by entering:
 

```
SETDefault !3a,!3b -PORT OWNeR=ppp
```
- 3 Enable IP routing by entering:
 

```
SETDefault -IP CONTROL=ro
```
- 4 Configure the console prompt by entering:
 

```
SETDefault -SYS NMPrompt="pim2 #"
```
- 5 Configure IP addresses by entering:
 

```
SETDefault !3a -IP NETaddr=192.168.1.2
SETDefault !3b -IP NETaddr=192.168.2.2
```
- 6 Enable MIP control by entering:
 

```
SETDefault -MIP CONTROL=e
```
- 7 Enable OSPF by entering:
 

```
SETDefault !3a,!3b -OSPF CONTROL=e
```
- 8 Enable PIM-SMv2 by entering:
 

```
SETDefault !3a,!3b -PIM CONTROL=sm
```
- 9 Configure the PIM-SM Candidate BootstrapRouter by entering:
 

```
SETDefault -PIM SMControl=cbsr
```
- 10 Add a Candidate BootstrapRouter address by entering:
 

```
ADD -PIM CandidateBSR 192.168.2.2
```

**Procedure for PIM3** To configure the bridge/router labeled PIM3, follow these steps:

- 1 Set the baud rate and enable the paths by entering:

```
SETDefault !3a,!3b -PATH BAud=256 CONTROL=e
```

- 2 Configure PPP links by entering:

```
SETDefault !3a,!3b -PORT OWNer=ppp
```

- 3 Enable IP routing by entering:

```
SETDefault -IP CONTROL=ro
```

- 4 Configure the console prompt by entering:

```
SETDefault -SYS NMPrompt="pim3 #"
```

- 5 Configure IP addresses by entering:

```
SETDefault !1 -IP NETaddr=192.168.30.1
```

```
SETDefault !3a -IP NETaddr=192.168.3.3
```

```
SETDefault !3b -IP NETaddr=192.168.2.3
```

- 6 Enable MIP control by entering:

```
SETDefault -MIP CONTROL=e
```

- 7 Enable OSPF by entering:

```
SETDefault !1,!3a,!3b -OSPF CONTROL=e
```

- 8 Enable PIM-SMv2 by entering:

```
SETDefault !1,!3a,!3b -PIM CONTROL=sm
```

**Procedure for PIM4** To configure the bridge/router labeled PIM4, follow these steps:

- 1 Set the baud rate and enable the paths by entering:

```
SETDefault !3a,!3b -PATH BAud=256 CONTROL=e
```

- 2 Configure PPP links by entering:

```
SETDefault !3a,!3b -PORT OWNer=ppp
```

- 3 Enable IP routing by entering:

```
SETDefault -IP CONTROL=ro
```

- 4 Configure the console prompt by entering:

```
SETDefault -SYS NMPrompt="pim4 #"
```

- 5 Configure IP addresses by entering:

```
SETDefault !1 -IP NETaddr=192.168.40.1
```

```
SETDefault !3a -IP NETaddr=192.168.3.4
```

```
SETDefault !3b -IP NETaddr=192.168.4.4
```

- 6 Enable MIP control by entering:

```
SETDefault -MIP CONTROL=e
```

- 7 Enable OSPF by entering:

```
SETDefault !1,!3a,!3b -OSPF CONTROL=e
```

- 8 Enable PIM-SMv2 by entering:

```
SETDefault !1,!3a,!3b -PIM CONTROL=sm
```

---

## How PIM-SM Works

This section gives a brief overview of the functioning of PIM-SM protocol.

### Multicast Routing Mechanisms

Multicast routing mechanisms such as Distance Vector Multicast Routing Protocol (DVMRP) and Multicast Open Shortest Path First (MOSPF) send data packets (in the case of DVMRP) or membership reports (in the case of MOSPF) periodically on many links that do not lead to receivers and senders. The periodic broadcasting of information by these protocols is required to identify the location of the interested receivers for a specific multicast session. This approach is useful in networks where bandwidth is plentiful, or when there are large number of senders and receivers for a multicast session. However, when senders and receivers to multicast sessions are distributed sparsely across a wide area, such schemes are not efficient; since they lead to wasted bandwidth on expensive WAN links and require the maintenance of “routing-state” on routers that are not on the forwarding tree for the multicast session.

Protocol Independent Multicast-Sparse Mode (PIM-SM) is a multicast routing protocol designed to resolve the above mentioned inadequacies with the “broadcast-and-prune” protocols like DVMRP and MOSPF. The term “protocol independent” comes from the fact that PIM can work with any unicast routing protocol. DVMRP maintains its own unicast routing table, and MOSPF depends on the OSPF link-state-database to build source based trees.

PIM-SM builds a per-group (or per multicast session) shared multicast distribution tree centered at a rendezvous point and requires receivers to explicitly join to this shared distribution tree before they receive data traffic. This approach ensures that bandwidth does not get wasted on links that do not have any downstream receivers. Since a “shared-tree” mechanism could result in suboptimal paths for data traffic from the sender to receivers, PIM-SM also supports the ability to switch to a source specific tree if the data traffic warrants it.

### Rendezvous Points

Rendezvous points (or RPs) in PIM-SM provide the mechanism for receivers of a multicast group to *meet* senders to the same group. rendezvous points are used by senders to announce their existence and by receivers to learn about new senders to a group. PIM-SM builds a per-group (or per multicast session) shared multicast distribution tree centered at a rendezvous point, and requires receivers to explicitly join to this shared distribution tree before they can receive data traffic. Designated routers (DRs) with directly connected sources encapsulate multicast data traffic in PIM control messages and unicast it to the rendezvous points until the rendezvous point tells the DR to start sending multicast natively.

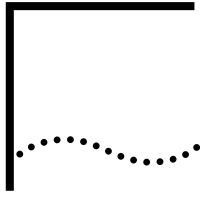
### Bootstrap Router

To obtain information about the rendezvous points, all routers within a PIM domain (a contiguous set of routers that all implement PIM-SM) collect Bootstrap messages. Bootstrap messages contain the list of active rendezvous points and the group prefixes served by each. The domain's bootstrap router is responsible for originating the Bootstrap messages, which are then forwarded hop-by-hop by PIM routers. A PIM-SM domain must have at least one router which is capable of functioning as a bootstrap router. When multiple candidate bootstrap routers exist in the PIM-SM domain, the bootstrap messages are used to carry out a dynamic bootstrap router election to determine the PIM domain's bootstrap router.

- Joining a Group** When a receiver wants to join a multicast session “G,” it conveys its membership information through the Internet Group Management Protocol (IGMP). This IGMP message is called an IGMP Join (or IGMP report). The PIM designated router on the LAN receives this IGMP report and looks up the associated rendezvous point for this group. After creating the appropriate multicast route entry, the DR sends a Join message via multicast (to the ALL-PIM-ROUTERS’ group, 224.0.0.13) on the RPF interface towards the rendezvous point. As this Join message propagates hop-by-hop to the rendezvous point each intermediate upstream router towards the rendezvous point creates or updates its multicast route entry for this multicast group. Thus a shared tree *rooted* at the rendezvous point gets created, for forwarding data traffic destined to the group G.
- Sending Data to a Group** When a sender “S” starts sending multicast data packets to a group, its DR initially delivers each packet to the rendezvous point for distribution down the rendezvous point-tree. The sender’s DR encapsulates each data packet in a PIM-Register message and unicasts it to the rendezvous point for that group. The rendezvous point decapsulates each register message and forwards the enclosed data packet natively to downstream members on the shared rendezvous point-tree.
- Switching from a Shared Tree to a Shortest Path Tree** If the data rate of the source warrants the use of a source-specific shortest path tree (SPT), the rendezvous point or PIM routers with local IGMP members may construct a new multicast route entry that is specific to the source, referred to as (S,G) state, and send periodic Join messages toward the source S (instead of the rendezvous point). When this (S,G) Join message reaches the DR directly connected to the source, all routers between the source’s DR and the router which initiated the (S,G) Join process possess (S,G) state to forward data packets destined for group G using a SPT. Once data traffic for group G starts arriving on the SPT, the router which initiates the (S,G) join prunes itself off the shared tree (since it now has a better path to the source).
- Leaving a Group** When the group has no more directly connected listeners (or receivers), the DR on the LAN gets notified via IGMP. If the DR has neither local members nor downstream receivers, it initiates the process to prune itself off the forwarding tree for this group. This pruning process is performed by sending PIM-Prune messages to the source or the rendezvous point, depending on whether the DR had (S,G) or (\*,G) state for this multicast group.
- PIM Packet Formats** The current version of PIM Sparse Mode is 2A II PIM control messages use the IP protocol number 103. . PIM control messages are either unicast (such as, Register messages from the PIM designated router to the rendezvous point), or multicast hop-by-hop (such as, Join/Prune and Assert to the ALL-PIM-ROUTERS multicast group (224.0.0.13)).







# CONFIGURING RSVP

This chapter describes the Resource Reservation Protocol (RSVP), which is used by multicast applications like video conferencing, multimedia, and virtual private network (VPN) network management. RSVP permits applications to request Quality of Service assurances from the network.

---

## What Is RSVP?

RSVP provides the ability to reserve resources for consistent data delivery for applications that need it. Data applications need a relatively small amount of bandwidth, but multimedia applications demand high bandwidth. With both types of applications using the same network, RSVP allows the multimedia applications to reserve the bandwidth they need to successfully complete their transmission.

RSVP provides network consistency for realtime traffic. Without this network consistency, real time traffic can experience information loss, jitter, loss of synchronization, and not enough bandwidth.

There are three RSVP participants: the sender, the network, and the receiver. There can be multiple senders and receivers. Each sender application periodically sends an RSVP Path message to a receiver for each data flow it originates. One piece of information provided in the Path message is the characteristics of the data traffic the application expects to generate. These characteristics are the data rate (bandwidth), the queue size, and the maximum packet size (MTU).

The Path message travels from a sender to receiver(s) along the same route(s) used by data packets. A bridge/router in the network that does not implement RSVP, routes the Path message through as if it were a data packet. An RSVP-capable bridge/router, processes the information in the Path message and uses it later in the reservation request message(s) sent back in the reverse direction to the sender.

The receiver application initiates reservation requests based on information it receives from the Path message. Each bridge/router that receives a reservation request message (Resv) reserves the requested bandwidth, if there is sufficient bandwidth, and sends the Resv message to its previous hop which is the next bridge/router in the route toward the sender. If there is not sufficient bandwidth, the Resv message goes no further and an error message is sent back to the receiver application. One other requirement for a successful reservation request is that, on a per flow basis, bandwidth greater than the user-configurable MaxFlowRate parameter cannot be requested.

## RSVP Configuration Example

This section describes a sample RSVP configuration example.

- 1 Configure and enable IP routing on LAN port !1 and Frame Relay virtual port !V1:

```
ADD !1 -IP NETaddr = <ipaddr1>
ADD !V1 -IP NETaddr = <ipaddr2>
SETDefault -IP CONTROL = ROUTe
```

- 2 Configure and enable IP routing protocol OSPF for unicast routing.
- 3 Configure and enable IP multicast routing protocol DVMRP and/or MOSPF:

```
SETDefault -MIP CONTROL = Enable
ADD !V1 -DVMRP Neighbor @dlci
SETDefault !1 -DVMRP CONTROL = Enable
SETDefault !V1 -DVMRP CONTROL = Enable
```

- 4 Configure the committed information rate of the Frame Relay virtual port:

```
SETDefault !V1 -FR CIR = <vcid>
```

- 5 Configure protocol reservation on the Frame Relay port:

```
ADD !V1 -PORT PROTOcolRsrv RSVP 60 # reserve 60% port bandwidth for
RSVP #
SETDefault !V1 -PORT QueueCONTROL = PROTOcolRsrv
```

This sample UI requests 60% of the available bandwidth for RSVP.

- 6 Configure and enable RSVP:

```
SETDefault -RSVP CONTROL = ENable
SETDefault !V1 -RSVP MaxFlowRate = 100 # limit perflow bandwidth to
100 bytes/sec #
```

For complete information on the commands and parameters, see the RSVP Service Parameters chapter in *Reference for Enterprise OS Software*.

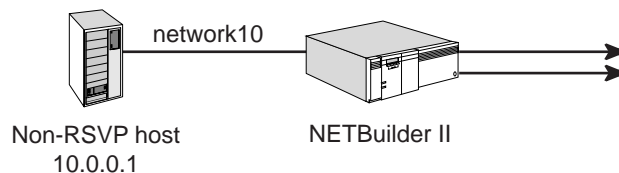
## RSVP Proxy Sender and Receiver

RSVP allows host applications to make bandwidth reservations in routers along the data path from senders to receivers. However, for "dumb" devices, such as IP telephone handsets, which are not connected to a PC or other host devices, reserving bandwidth using RSVP would not be possible. RSVP Proxy sender and receiver solve this problem by emulating RSVP senders or receivers on behalf of these devices.

The following examples demonstrate how this feature works.

### Proxy Sender: Unicast Destination and One Sender Port

**Figure 85** RSVP Proxy Sender with a Unicast Destination and One Sender Port



To configure the example shown in Figure 85, enter:

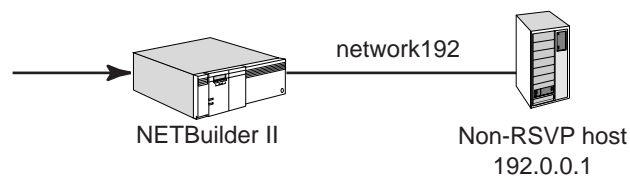
```
ADD -RSVP ProxySENDER EX1_PS SESSION 192.0.0.1/2500 UDP SENDER
10.0.0.1/3000 RATE 2000 1000 TimeOut 40
```

This ADD command emulates a proxy sender for the non-RSVP host whose IP address is 10.0.0.1. Data transmitted by the non-RSVP host on UDP port number 3000 and destined for unicast address 192.0.0.1 port 2500 initiate an RSVP session. The RATE value also specifies the characteristics of the data traffic expected to be generated by the sender host of 2000 bytes per second with burst size of 1000 bytes.

A TimeOut period of 40 seconds is specified which would cause the RSVP session to be torn down should the sender stop transmitting data for that length of time. This parameter is optional; if not specified, a 300 second idle out period is assumed. A specification of 0 disables the timer.

### Proxy Receiver: Unicast Destination and One Sender Port

**Figure 86** RSVP Proxy Receiver with a Unicast Destination and One Sender



To configure the example shown in Figure 86, enter:

```
ADD -RSVP ProxyREceiver EX1_PR SESSion 192.0.0.1/2500 UDP SENDer
10.0.0.1/3000 RATE 2000 1000 STYLE FixedFilter
```

This ADD command emulates a RSVP receiver for non-RSVP host, 192.0.0.1. A RSVP path message received for session 192.0.0.1 port 2500 with UDP protocol ID causes an RSVP reservation request message to be sent by the NETBuilder on behalf of the non-RSVP host. As specified by the RATE parameter in the ADD parameter, the request is for a bandwidth of 2000 bytes per second and burst size of 1000 bytes for data transmitted by sender 10.0.0.1 on port 3000. The requested bandwidth is for the exclusive use of data sent by 10.0.0.1 on port 3000, as specified by the FixedFilter reservation style.

### Proxy Sender: Multicast Destination with a Range of Sender Ports

To configure proxy sender for a multicast destination with a range of sender ports, enter:

```
ADD -RSVP ProxySENDER EX1_PSM SESSion 239.0.0.2/2500 TCP
SENDER10.0.0.1/3000-3017 RATE 2000 1000 TimeOut 40
```

This ADD command is the same as the ADD command for a unicast session, specified above, except for the multicast destination address in the SESSion parameter, the TCP protocol ID and the range of sender ports.

Although not shown in the ADD command above, the SESSion parameter may also contain a port range, as in the sender port range, which would initiate multiple RSVP sessions, one for each port number within the range.

### Proxy Receiver: Multicast Destination with a Range of Sender Ports

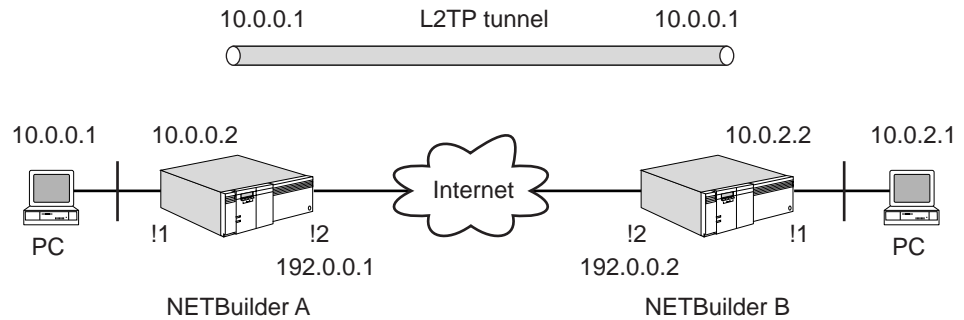
To configure proxy receiver for a multicast destination with a range of sender ports, enter:

```
ADD -RSVP ProxyREceiverer EX1_PRM SESSion 239.0.0.1/2500 TCP SENDer
10.0.0.1/3000-3017 RATE 2000 1000 STYLE SharedExplicit
```

The values for this command are the same as the unicast proxy receiver ADD command except for the multicast destination address, the TCP protocol ID, the sender port range and the SharedExplicit reservation style. A reserved bandwidth of 2000 bytes per second and 1000 byte burst size is requested and shared by data transmitted by the sender, 10.0.0.1, on ports 3000 to 3017 inclusive.

### Sample RSVP Configuration with L2TP Tunnel

Figure 87 RSVP for L2TP Tunnel



In this topology, there is an L2TP VLL between NETBuilder A and NETBuilder B. This configuration reserves a large percentage of the tunnel's bandwidth capacity for L2TP traffic, with the rest of the tunnel available for Internet access.

In this case, both NETBuilder bridge/routers are RSVP-aware routers, but the host PC's are not. L2TP uses UDP port 1701 to send packets. Also, the outer IP addresses of the WAN link (192.0.0.x) are used as sender and destination addresses.

Assuming the tunnel bandwidth is 64K bit/s and you want to reserve 80% for RSVP/L2TP, and the average traffic on the tunnel is 10K bit/s full duplex.

To create the topology illustrated in Figure 87, follow these steps.

- 1 Setup L2TP VLL. (See the Configuring L2Tunnel Connections chapter of *Using Enterprise OS Software*.)
- 2 On NETBuilder A enter:

```
SETDefault !2 -Port qcont = protr
ADD !2 -Port protr RSVP 80
```

This sets up the WAN port to use protocol reservation for queueing and to reserve 80% of the port's bandwidth for RSVP.

- 3 Add a proxy sender to NETBuilder A by entering:

```
ADD -RSVP psend L2TPA sess 192.0.0.2/1701 UDP sender 192.0.0.1/1701 rate
1000 6400
```



*The bandwidth is reserved in bytes.*

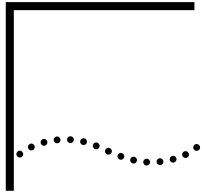
Data transmitted by the PC host to destination address 192.0.0.2 port 1701 causes a RSVP PATH message to be sent to NETBuilder B.

- 4 Add a proxy receiver on NETBuilder B by entering:

```
ADD -RSVP prec L2TPB sess 192.0.0.2/1701 UDP sender 192.0.0.1/1701 rate
1000 6400 style FixedFilter
```

A RSVP PATH message received by NETBuilder B for the session from sender 192.0.0.1 on port 1701 causes a RSVP RESV message to be generated which requests the amount of bandwidth as specified by the RATE value in the ADD command.





# CONFIGURING DHCP

This chapter describes the Dynamic Host Configuration Protocol (DHCP), which allows the NETBuilder bridge/router to maintain a pool of IP addresses available to clients upon request. DHCP reduces the complexity of configuring computers for a Transmission Control Protocol/Internet Protocol (TCP/IP) network.

---

## Configuring DHCP

DHCP dynamically assigns an IP address to the requesting client for a specified duration of time. When the client no longer needs this IP address, or when the assigned duration expires, the address is returned to the pool. DHCP eliminates manually assigning an address to each client and provides efficient use of your finite number of IP addresses, especially for remote access clients.

When your TCP/IP environment already exists, you can use the existing IP address scenario and set up a DHCP address pool using the network mask that matches the primary network address. A DHCP profile can be defined to supply configuration information as well as addressing to the DHCP client. Profile information can include gateway, DNS servers, WINS name server, Domain Name, lease times, renew timer, and rebind timer information.

AddressPool and DefAddrPool are LAN only. RasAddrPool is for RAS only.

**Procedure** To configure DHCP on the NETBuilder bridge /router, follow these steps:

- 1 Enable DHCP on the NETBuilder II bridge/router. Enter:

```
SETDefault !1 -DHCP CONTROL=Enable
```

- 2 Set up the DHCP IP address pool. Enter:

```
SETDefault !1 -DHCP CONTROL= AddressPool
ADD !1 -DHCP AddressPool 129.000.00.00 - 129.000.00.00 !P1
```

This IP address range is now assigned to the DHCP address pool and is available for all client requests.

- 3 Map a static address, which associates a fixed IP address with the server MAC address. This step is optional - this step is only used for the remote RAS client.

```
ADD !1 -DHCP StaticAddress %080002188E57 129.000.00.00 !p1
```

- 4 Create a RAS address pool for RAS clients:

```
ADD !1 -DHCP RasAddressPool 129.000.00.000 - 129.000.00.000 !P4
```

This setup allows remote access clients to acquire an IP address when they dial in for the duration of the session. DHCP manages a separate RAS IP address pool, associated with a configuration set, for the RAS clients. The allocation of a valid address and configuration is managed locally in the NETBuilder bridge/router.

DHCP configuration consists of two major steps: configuring the address pools and selecting the configurations options.

---

## Configuring Address Pools

DHCP manages its available address pools through AddressPool and DefAddressPool for external DHCP clients. DHCP also manages another address database called RasAddressPool for its internal RAS clients. DHCP also supplies a set of configurations to its client, called the DHCP profile.

Supported DHCP configurations include:

- ProfDefGateWay (DHCP option 3)
- ProfDNS (DHCP option 6)
- ProfDomainName (DHCP option 15)
- ProfLEASE (DHCP option 51)
- ProfNetBiosNs (DHCP option 44)
- ProfReBindTimer (DHCP option 59)
- ProfReNewTimer (DHCP option 58)
- ProfSubnetMask (DHCP option 1).
- For complete information on the commands and parameters discussed in this section, see the DHCP Service Chapter Parameters in *Reference for Enterprise OS Software*.

---

## Enabling DHCP Service

CONTRol is a parameter used to enable the DHCP Service. CONTRol is a per-port parameter that contains three control elements:

- Enabled/Disabled
- AddressPool/DefAddressPool
- ICMPCheck/NoICMPCheck

For DHCP to be operational on a port, the associated port CONTRol must be enabled. For each port, DHCP maintains an active address pool, either AddressPool or DefAddressPool, from which to allocate an available address for its external client when requested. You can choose either AddressPool or DefAddressPool as the active address pool on a port through the AddressPool/DefAddressPool bit from the CONTRol parameter. The IcmpCheck/NoIcmpCheck in the CONTRol parameter determine whether DHCP will execute the ICMP ECHO checking before it assigns an address to its client. For example, when you enter:

```
SETDefault !2 -DHCP CONTRol = (Enabled,AddressPool,IcmpCheck)
```

DHCP operation is enabled on port 2. AddressPool is the active address pool and DHCP retrieves an available address and assigns it to the client when requested. DHCP executes the ICMP ECHO checking to make sure no other network device is using this address, before DHCP assigns it to the client.

---

## Configuring Address Pools

When CONTRol is configured with AddressPool, the AddressPool is considered the active address pool. When CONTRol is configured with DefAddressPool, the DefAddressPool is considered as active address pool. At any given time only one



address pool (either AddressPool or DefAddressPool) can be configured as an active address pool on a particular port.

**AddressPool** Before DHCP assigns an IP address to its external client, it checks the validity of the particular network. Currently, DHCP only operates on the network number that matches the primary NETaddr (-IP) and the network mask. For example, when you enter:

```
ADD !2 -DHCP AddressPool 129.213.201.152 - 129.213.201.173 !P1
```

The address block of (129.213.201.152 - 129.213.201.173) is added with the associated profile !P1 into the AddressPool of port 2.

When you enter:

```
ADD !2 -DHCP AddressPool 129.213.201.185 - 129.213.201.193 !P1
```

The address block of (129.213.201.185 - 129.213.201.193) is added with the associated profile !P1 into the AddressPool of port 2.

Each address block is configured, optionally associated with a DHCP profile which contains a set of DHCP options to offer the client when a member IP address is selected. If no <profileid> is configured with an address block, the P1 profile is selected as the default profile.

It is valid to have multiple address blocks configured into an address pool particular port.

It is important that when multiple address blocks are configured, the associated profile of each address block should use the same profile set, so to keep DHCP a consistent configuration allocation when addresses are selected from different address blocks.

**DefAddressPool** DefAddressPool is generated by the system and cannot be modified. It can only be viewed by DefAddressPool parameter. The DefAddressPool is generated by the system on a LAN interface. The DefAddressPool is always associated with the profile P1.

### Procedure

To display the system-generated DefAddressPool on all LAN interfaces, enter:

```
[18]DPE # SHOW !* -DHCP DefAddressPool
```

This information appears on the screen:

```
-----Default Address Pool-----
Port Default Address Pool Profile
0 10.0.0.50 - 10.0.255.254 P1
1 10.1.0.50 - 10.1.255.254 P1
2 10.2.0.50 - 10.2.255.254 P1
3 10.3.0.50 - 10.3.255.254 P1
3B 10.11.0.50 - 10.11.255.254 P1
```

## Procedure

To display the system-generated DefAddressPool only on those LAN interfaces that have the CONTrol configured with Enabled, enter:

```
[19]DPE # SHow -DHCP DefAddressPool
```

The following information is displayed:

```
-----Default Address Pool-----
Port Default Address Pool Profile
2 10.2.0.50 - 10.2.255.254 P1
```

When DefAddressPool is selected as the active address pool, the network number on that port (-IP NETAddr) must be configured to match the network number of DefAddressPool. In the previous example, DefAddressPool on port !2 was automatically generated as (10.2.0.50 - 10.2.255.254). Now, you must configure the NETAddr to match the networks. For example, enter:

```
SETDefault !2 -IP NETAddr = 10.2.0.1 255.255.0.0
```

This can be configured automatically using Quick Step VPN. For complete information on the commands and parameters discussed in this section, see the DHCP Service Parameters chapter in *Reference for Enterprise OS Software*.

---

## Configuring the DHCP Options

A DHCP profile contains a set of network configurations attributes that DHCP offers (with IP address) to its clients. One default profile P1 is provided by the system. You can configure its contents.

### Procedure

To display the contents of the P1 default profile shipped with Enterprise OS software, enter:

```
[32]DPE # SHow -DHCP PProfile !P1
```

The following information is displayed:

```
Profile: 1
ProfDefGateWay = (AUTO)
ProfDNS = (AUTO) 129.213.128.98 128.9.0.107 198.41.0.4
192.33.4.12
 ProfDomainName = (NONE)
 ProfLEASE = 28800 (secs)
 ProfNetBiosNs = (NONE)
 ProfReBindTimer = 21600 (secs)
 ProfReNewTimer = 7200 (secs)
 ProfSubnetMask = (AUTO)
```

The contents of profile P1 is the default profile. This profile is also used as a template when adding a new profile. The details for each field (option) in a profile is described in the DHCP Service Parameter chapter in *Reference for Enterprise OS Software*.

Some profile fields support AUTO. AUTO in each option field means automatically derived by the system. Some profile fields support NONE. NONE in each option field means not available, and that associated option is not offered to the client when addresses are assigned to the client.

For complete information on the commands and parameters discussed in this section, see the DHCP Service Parameters chapter in *Reference for Enterprise OS Software*.

## Creating a New DHCP Profile

You can create up to 32 custom DHCP profiles and change the contents of each profile. The contents of profile !p1 are provided by the Enterprise OS software. You can change each option value of the profile p1. The profile p1 is used as the default profile throughout the DHCP service.

**Procedures** The following commands can be issued to perform various functions.

- 1 To change the ProfDefGateWay to 129.213.128.122 in an existing DHCP profile, enter:

```
SETDefault !P1 -DHCP ProfDefGateWay = 129.213.128.122
```

- 2 To eliminate ProfDefGateWay as an option in DHCP profile P1 when DHCP assigns an address, enter:

```
SETDefault !P1 -DHCP ProfDefGateWay = NONE
```

- 3 To automatically derive the ProfDefGateWay in DHCP profile P1, when profile P1 is used when assigning an address, the system offers ProfDefGateWay with some value derived by the system, enter:

```
SETDefault !P1 -DHCP ProfDefGateWay = AUTO
```

For more information about ProfDefGateWay, see the DHCP Service Parameters chapter.

- 4 To specify 129.213.128.16 and 129.213.128.2 as the ProfDNS in DHCP profile P1, enter:

```
SETDefault !P1 -DHCP ProfDNS = 129.213.128.16 129.213.128.2
```

- 5 To eliminate ProfDNS in DHCP profile P1 when profile P1 is used to assign an address, enter:

```
SETDefault !P1 -DHCP ProfDNS = NONE
```

- 6 To automatically derive the ProfDNS in DHCP profile P1, enter:

```
SETDefault !P1 -DHCP ProfDNS = AUTO
```

The four values will be set as AUTO: 129.213.128.98, 128.9.0.107, 198.41.0.4, and 192.33.4.12 will be set as the values. When profile P1 is used to assign an address, the system offers ProfDefGateWay with these values.

- 1 To specify "ewd.3com.com" as the ProfDomainName in DHCP profile P1, enter:

```
SETDefault !P1 -DHCP ProfDomainName = "ewd.3com.com"
```

- 2 To eliminate ProfDomainName as an option in the DHCP profile P1 when profile P1 is used, enter:

```
SETDefault !P1 -DHCP ProfDomainName = NONE
```

- 3 To eliminate ProfDomainName in DHCP profile P1, enter:

```
SETDefault !P1 -DHCP ProfDomainName = ""
```

- 4 The empty string "" is equivalent to the NONE. ProfDomainName is not configured and not offered to its client.

- 5 To specify 129.213.128.36 and 129.213.128.22 as the ProfNetBiosNs in DHCP profile P1, enter:

```
SETDefault !P1 -DHCP ProfNetBiosNs = 129.213.128.36 129.213.128.22
```

- 6 To eliminate ProfNetBiosNs as an option in DHCP profile P1, enter:

```
SETDefault !P1 -DHCP ProfNetBiosNs = NONE
```

- 7 To specify the ProfLEASE value in DHCP profile P1 as 7200 (seconds), enter:

```
SETDefault !P1 -DHCP ProfLEASE = 7200
```

- 8 To specify the ProfReNewTimer value in DHCP profile P1 as 3600 (seconds), enter:

```
SETDefault !P1 -DHCP ProfReNewTimer = 3600
```

The ProfReNewTimer indicates the renew time that DHCP client the uses to start its renewal cycle.

- 1 To specify the ProfReBindTimer value in DHCP profile P1 as 5400 (seconds), enter:

```
SETDefault !P1 -DHCP ProfReBindTimer = 5400
```

- 2 The ProfReBindTimer specifies the renew time that DHCP client uses to start its rebind cycle. The ProfReNewTimer should be less than ProfReBindTimer and less than ProfLEASE. For example, configure values with the following relationship:

```
ProfReNewTimer = 1/2 * ProfLEASE
ProfReBindTimer = 3/4 * ProfLEASE
```

- 3 To specify 255.255.255.0 as the ProfSubnetMask in DHCP profile P1, enter:

```
SETDefault !P1 -DHCP ProfSubnetMask = 255.255.255.0
```

- 4 To eliminate ProfSubnetMask in DHCP profile P1, enter:

```
SETDefault !P1 -DHCP ProfSubnetMask = NONE
```

- 5 To automatically derive the ProfSubnetMask in DHCP profile P1 so that when profile P1 is used when assigning an address, the system offers ProfDefGateWay with a value derived by the system, enter:

```
SETDefault !P1 -DHCP ProfSubnetMask = AUTO
```

For complete information on the commands and parameters discussed in this section, see the DHCP Service Parameters chapter in *Reference for Enterprise OS Software*.

## Configuring DHCP Profiles

DHCP supports a maximum of 32 profiles with the profile P1 provided by the system. You can create up to 32 custom DHCP profiles and change the contents of each profile. When you add a new profile, the NETBuilder II bridge/router copies the contents of profile p1 into the new profile as template contents. You can change each option of a custom profile after it is created.

- 1 You can add a new profile (P2 up to P32) to be associated with the address assignments. For example, enter:

```
ADD -DHCP PProfile P7
```

- 2 You can delete a profile (P2 up to P32) except default profile p1. For example, enter:

```
DElete -DHCP PProfile P7
```

- 3 You can display how many DHCP profiles are currently configured. For example, enter:

```
SHoW -DHCP PRofile
```

The following display appears:

```
[5]DPE # SHoW -DHCP PRofile
Profile P1
Profile P7
```

- 4 You can display the contents of a specific DHCP profiles. For example, enter:

```
SHoW -DHCP PRofile P7
```

The following display appears:

```
[6]DPE # SHoW -DHCP PRofile P7
Profile: 7
ProfDefGateWay = (AUTO)
ProfDNS = (AUTO) 129.213.128.98 128.9.0.107 198.41.0.4
192.33.4.12
ProfDomainName = (NONE)
ProfLEASE = 28800 (secs)
ProfNetBiosNs = (NONE)
ProfReBindTimer = 21600 (secs)
ProfReNewTimer = 7200 (secs)
ProfSubnetMask = (AUTO)
```

- 5 You can map ProfClassIdent with a custom profile. For example, enter:

```
ADD !2 -DHCP ProfClassIdent EngHost P4
```

This profile overrides the associated profile when the address is selected and the client identifies itself with a Class Identifier option. The following information appears:

```
[49]DPE # SHoW !* -DHCP ProfClassIdent
```

```
-----Class Identifier Profile Mapping Table-----
Port Class Identifier Profile
2 DavidsHost P4
5 DCH P3
```

```
[50]DPE # SHoW -DHCP ProfClassIdent
```

```
-----Class Identifier Profile Mapping Table-----
Port Class Identifier Profile
2 DavidsHost P4
```

- 6 You can assign a static address: For example, enter:

```
ADD !2 -DHCP StaticAddress %0020AF735FB1 129.213.201.167
```

StaticAddress allows you to maintain a special static mapping with the client's MAC address, By default, each dynamic address is selected from the active address except when a static address mapping is found.

The following information is displayed:

```
[27]DPE # SHoW !* -DHCP StaticAddress
```

```
-----Static Address Mapping Table-----
Port Host Address IP address Profile
```

```

2 %0020AF735FB1 129.213.201.167 P1
5 %0020AF4165EE 13.13.13.13 P1

```

With these configurations, a client with MAC address of %0020AF735FB1 requesting received from port !2 will always get IP address of 129.213.201.167 allocated. A client with MAC of %0020AF4165EE requesting received from port !5 will always get the 13.13.13.13 IP address allocated.

- 7 You can delete a static address: For example, enter:

```
DELeTe !5 -DHCP StaticAddress %0020AF4165EE 13.13.13.13
```

This command deletes this static mapping from the table.

- 8 You can maintain a internal address pool to be used by the Remote Access Server (RAS) client: For example, enter:

```
ADD !1 -DHCP RasAddressPool 20.20.20.25 - 20.20.20.55 P1
```

When the RAS client makes an address request to DHCP service, it is this address pool that DHCP uses to allocate the RAS address. This command adds the address block (20.20.20.25 - 20.20.20.55) associated with profile P1 into the RasAddressPool of port !1.

- 9 You can delete the address block. For example, enter:

```
DELeTe !1 -DHCP RasAddressPool 20.20.20.25 - 20.20.20.55
```

This command deletes the address block (20.20.20.25 - 20.20.20.55) from !2 RAS.

- 10 You can display the log. For example, enter:

```
SHow -DHCP Log
[59]DPE # SHow -DHCP Log
Log = (NoSyslog,NoConsole)
[60]DPE #

```

- 11 You can turn on the DHCP syslog feature. For example, enter:

```
SETDefault -DHCP Log = Syslog
```

When Log is configured with Syslog, key operational information is sent to the system log server.

The sys log server must be configured through the AUDIT LOG service, before DHCP can send system log messages.

- 12 You can turn on the DHCP 'Console' feature. For example, enter:

```
SETDefault -DHCP Log = Console
```

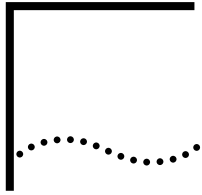
## Logging to the Console

When Log is configured with Console, key operational information is sent to the console. The Console logging shows more detail information than syslog









# CONFIGURING L2TUNNEL CONNECTIONS

This chapter describes how to configure a NETBuilder bridge/router as a tunnel terminator packet processor, how to configure a bridge/router as a tunnel initiator/terminator in a router-to-router configuration, and how to configure virtual leased lines with the Point-to-Point Tunneling Protocol (PPTP) and Layer 2 Tunneling Protocol (L2TP).

PPTP/L2TP defines a method for transferring Point-to-Point Protocol (PPP) datagrams through a tunnel over IP. Tunneling PPP does not change PPP but provides a vehicle by which PPP data units (PDUs) can be carried between two peers: a line server (LS) and a packet processor (PP). The LS-PP pair defines the endpoints of a PPTP/L2TP connection.

A PPTP/L2TP connection is defined by two parallel components: a control connection and a data pipe. Both operate between the same LS-PP pair. For PPTP, the control connection operates over TCP and passes call control and management packets over the TCP session. The data pipe operates over IP to transfer data packets encapsulated using Generic Routing Encapsulation Protocol Version 2 (GRE V2). For L2TP, both the control connection and the data pipe operate over the UDP session.

When the NETBuilder bridge/router adopts the PPTP/L2TP protocol, it functions as a packet processor. You can configure PPTP/L2TP tunnel connections between a NETBuilder bridge/router and a 3Com AccessBuilder® server (PPTP only) or a Total Control™ hub (acting as a line server). In this scenario, the NETBuilder bridge/router is acting as a tunnel terminator, which only receives inbound calls. A NETBuilder bridge/router can also receive inbound calls from VPN-capable RAS clients (Windows 98/NT) to provide remote access services.

In addition, a NETBuilder bridge/router expands the use of PPTP/L2TP so a tunnel can be established between two peer NETBuilder bridge/routers. In this scenario, the NETBuilder bridge/router is able to issue outbound calls so either side can be the tunnel initiator or tunnel terminator. This is a router-to-router configuration, and both peers play the same role.

---

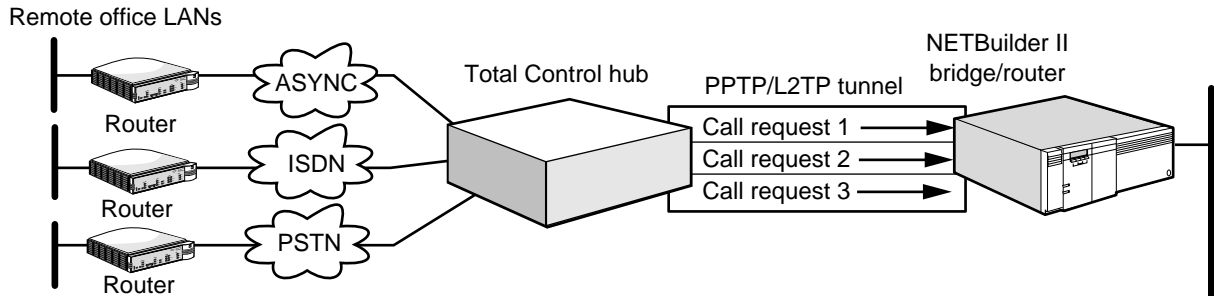
## Configuring a NETBuilder Bridge/Router as a Tunnel Terminator (PP)

When using a NETBuilder bridge/router as a packet processor, you can choose the following hubs as a line server:

- AccessBuilder hub 4000 and 5000 models version 6.2 and above (PPTP only)
- AccessBuilder hub 8000 model version 3.6 and above (PPTP only)
- Total Control hub with NetServer Card version 3.4 and above, or with HyperArc card

In Figure 88, the PPTP/L2TP tunnel connections are configured between a NETBuilder bridge/router and a Total Control hub.

**Figure 88** PPTP/L2TP Tunnel Connections Between a Bridge/Router and Total Control Hub



By default, the LS with any IP address will be able to connect to the bridge/router using PPTP/L2TP. Flow control for all PPTP/L2TP sessions is disabled by default. If you need to enable flow control or if you want to restrict which line servers can have PPTP/L2TP connections with the bridge/router, you can configure an access list. After you have configured an access list, the bridge/router accepts PPTP/L2TP connections only from an LS whose IP address has been specified. The flow control for each PPTP connection is configurable.



*The configuration for L2TP is identical to that for PPTP, except that L2TP has its own flow control mechanism embedded in the L2TP protocol. No flow control configuration is required for L2TP tunnels. You also need to define L2TPLocalUser and L2TPRemoteUser on both routers.*

To configure a NETBuilder bridge/router as a tunnel terminator, follow these steps:

- 1 Enable the L2Tunnel service by entering:

```
SETDefault -L2Tunnel CONTROL = Enabled Protocol = PPTP
```

or

```
SETDefault -L2Tunnel CONTROL = Enabled Protocol = L2TP
```

or

```
SETDefault -L2Tunnel CONTROL = Enabled Protocol = ALL
```

- 2 Configure the access list if you want to restrict the incoming peer tunnel initiators or if you want to enable the PPTP flow control using:

```
ADD -L2Tunnel AccessList <IP Address> [<Subnet Mask>] [FlowControl=Enabled | Disabled] [Protocol = PPTP | L2TP | ALL]
```

Use this command you configure the IP address or IP address range of acceptable tunnel initiators. By default, FlowControl is disabled.

For example, to enable a line server with an IP address of 129.213.48.6 to have a PPTP connection with the bridge/router, enter:

```
ADD -L2Tunnel AccessList 129.213.48.6 255.255.255.255 Protocol = PPTP
```

You can also create an access list for a group of IP addresses representing an entire subnet. For example, to create an access list for an entire subnet 129.213.48.0 for both PPTP and L2TP, enter:

```
ADD -L2Tunnel AccessList 129.213.48.0 255.255.255.0 Protocol = ALL
```

- 3 Repeat step 2 for each IP address or IP address range you want to add to the access list. To display the access list being configured, enter:

```
SHow -L2Tunnel AccessList
```

To remove a previously-configured access list entry, use:

```
DELeTe -L2Tunnel AccessList <IP Address>
```

- 4 If not yet configured, configure either SysCallerID or AuthRemoteUser of the virtual port you have chosen to be bound to the incoming virtual path using:

```
ADD !<port> -PORT VirtualPort SCID "<SysCallerID>"
```

Or

```
ADD !<port> -PORT VirtualPort PPP
```

```
ADD !<port> -PPP AuthRemoteUser ("<userid>", "<password>")
```

For example, if you want to create virtual port !V1 to get connected to the LS box LS1, you should configure something like:

```
ADD !V1 -PORT VirtualPort SCID"LS1"
```

Or

```
ADD !V1 -PORT VirtualPort PPP
```

```
ADD !V1 -PPP AuthRemoteUser ("LS1", "LS1PW")
```

- 5 If a SysCallerID is used in step 4, configure SysCallerID of the bridge/router using:

```
SETDefault -SYS SysCallerID="<string>"
```

For example, if the SysCallerID for this bridge/router is "NB", use:

```
SETDefault -SYS SysCallerID="NB"
```

If AuthRemoteUser is used in Step 4, configure local user names and passwords for the port that is going to connect to the peer using:

```
SETDefault !<port> -PPP AuthLocalUser ("userid", "password")
```

For example, if the user name and password for PPP negotiation are "NB" and "NBPW", enter:

```
SETDefault !V1 -PPP AuthLocalUser=("NB","NBPW")
```



See the *L2Tunnel Service Parameters chapter in Reference for Enterprise OS Software* for additional parameters that you can optionally use to configure the L2TP connection.

- 6 To display statistics for all pptp connections that are currently active, enter:

```
SHow -L2Tunnel pptpSTATS
```

- 7 To display information about the state of each PPTP connection, enter:

```
SHow -L2Tunnel pptpSTATUS
```

- 8 To display information about the state and statistics of L2TP connections, enter:

```
SHow -L2Tunnel L2TPTunnels
```

```
SHow -L2Tunnel L2TPStats
```

- 9 If L2TP is used as the tunneling protocol, set the L2TP local and remote user name of the NETBuilder bridge/router using:

```
SETDefault -L2T L2TPLocalUser = ("userid", "Password")
```

```
ADD -L2T L2TPRemoteUser = ("userid", "Password")
```

The assigned L2TPLocalUser name will be used as the “Host Name” during tunnel establishment which is required by the L2TP protocol.



*11.1 software requires you to set up the system name for the L2TP tunnel to be established. This is not required in 11.2 and later releases. In 11.2, the L2TPLocalUser name not the system name is used for the host name.*

---

### Configuring a NETBuilder Bridge/Router as a Tunnel Initiator/Terminator (Router-to-Router)

When two NETBuilder bridge/routers are used in a router-to-router configuration, either NETBuilder bridge/router can issue dial commands. When a bridge/router issues a dial command, an outgoing call request message is sent to the peer. The peer responds by sending back an outgoing call reply message. A session is established within a PPTP/L2TP tunnel. (This setup mechanism is unlike the LS-PP scenario where incoming call messages are exchanged.)

To configure a NETBuilder bridge/router as a tunnel initiator/terminator, follow these steps:

- 1 Repeat steps 1 through 5 of the procedure “Configuring a NETBuilder Bridge/Router as a Tunnel Terminator (PP)” earlier in this chapter.
- 2 The DialNoList contains the IP address of the physical interface of the peer bridge/router. Configure a DialNoList for each dial out virtual port using:

```
ADD !<port> -PORT DialNoList "<@IP Address>" Type=PPTP
or
ADD !<port> -PORT DialNoList "<@IP Address>" Type=L2TP
```

For example, if the IP address of the physical interface of the peer bridge/router is 129.213.48.6 and it is intended to use PPTP tunneling and to connect it through virtual port !V1, enter:

```
ADD !V1 -PORT DialNoList "@129.213.48.6" Type=PPTP
```

- 3 To establish the PPTP connection, either NETBuilder bridge/router must issue a dial command using:

```
Dial !<port>
```

For example, to dial from virtual port !V1, enter:

```
Dial !V1
```

If the physical interface is a dial-up line, you need to configure your communication resources to use your dial-up lines before issuing the Dial command. See the Configuring Port Bandwidth Management chapter in *Using Enterprise OS Software* for information about configuring communication resources for dial-up use.

---

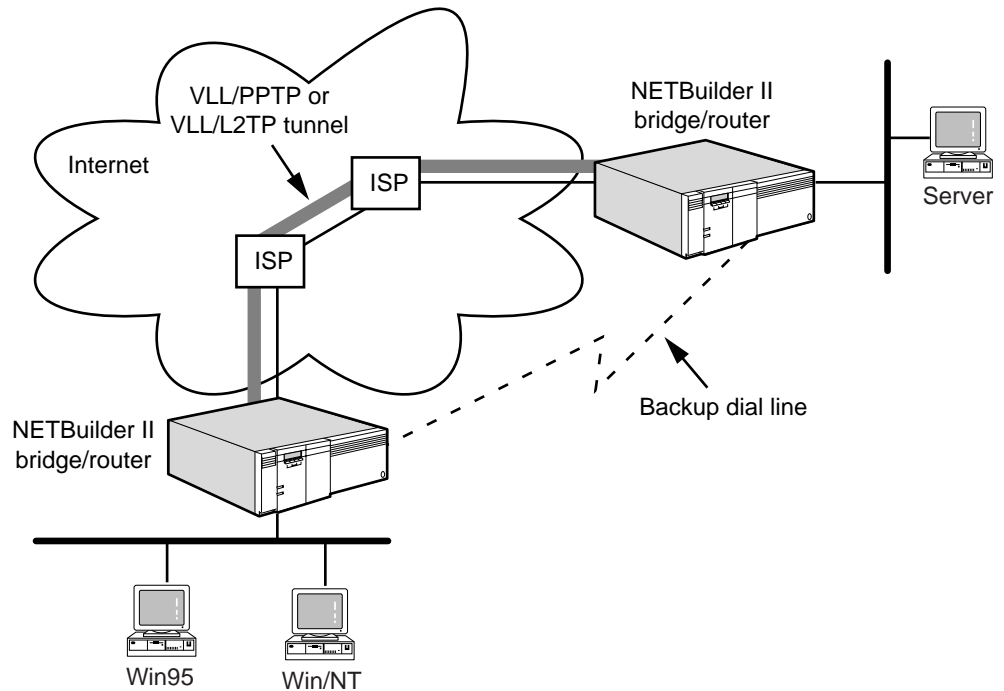
### Configuring Virtual Leased Line over PPTP/L2TP

When configuring a NETBuilder bridge/router as a tunnel initiator/terminator (router-to-router scenario), the Dial command establishes the tunnel. In this scenario, tunnel usage may be disrupted due to heavy traffic or poor line quality. In reliability sensitive environments, you may choose to configure the PPTP/L2TP tunnel to operate as a leased line.

If the tunnel goes down, the environment can be preconfigured so that a backup line can be dialed as shown in Figure 89. All traffic is then sent through the backup line. While the backup line is in use, the bridge/router tries to reestablish the PPTP/L2TP tunnel. When the tunnel is reestablished, all the traffic is switched

from the backup line to the tunnel. The backup line is then automatically torn down.

**Figure 89** Virtual Leased Line and Backup Connection



The PPTP/L2TP virtual leased line is brought up automatically after it has been configured. Parameters must be configured on both bridge/routers in order to bring up the line.

To configure a PPTP virtual leased line, follow these steps:

- 1 If not enabled, enable the L2Tunnel service and choose desired protocol by entering:

```
SETDefault -L2Tunnel CONTROL = Enabled
```

- 2 Add a virtual leased line using:

```
ADD -L2Tunnel VLeasedLine < IP Address>
```

Where <IP Address> is the IP address of the physical port of the peer.

On the peer router, there could be more than one physical port configured with different IP addresses. You should choose the IP address of the physical port through which the tunnel/TCP session will be established. For example, if you need to establish a virtual leased line over PPTP to the peer with physical port IP address 129.213.48.6, enter:

```
ADD -L2Tunnel VLeasedLine 129.213.48.6
```

In some environments, the IP addresses of the remote peers may not be known beforehand. For instance, the remote peer is located across the internet and its IP address is assigned dynamically by an Internet Service Provider (ISP). There is no way that you can preconfigure the IP address of the peer's physical port in step 2. In this case, you can configure a VLeasedLine IP address entry of 0.0.0.0, meaning that any IP address from a remote peer will be accepted. There should be one

0.0.0.0 IP address entry for each incoming remote peer with an unknown IP address.

- 3 To configure the backup line for virtual port under which PPTP is running, use:

```
SETDefault !<port> -PORT DialCONTROL = DisasterRcvry
```

For example, if virtual port !V1 is intended to configured a backup line, enter:

```
SETDefault !V1 -PORT DialCONTROL = DisasterRcvry
```

- 4 Add the dial number for the backup line, using:

```
ADD !<port> DialNoList "<phone-no>" [Type = Modem | Bri | Sw56 | WE]
```

If the backup line is going from port V1 through modem to the peer's physical port with dialed phone number 9241234, it should be configured as:

```
ADD !V1 DialNoList "9241234" Type=Modem
```

- 5 If not yet configured, configure either SysCallerID or AuthRemoteUser of the virtual port that you choose to be bound to the incoming virtual path using:

```
ADD !<port> -PORT VirtualPort SCID"<sysCallerID>"
```

or

```
ADD !<port> - PORT VirtualPort PPP
```

```
ADD !<port> -PPP AuthRemoteUser ("<userid>", "<password>")
```

For example, if you want to create virtual port !V1 to get connected to the LS box LS1, you should configure something like:

```
ADD !V1 -PORT VirtualPort SCID"LS1"
```

or

```
ADD !V1 -PORT VirtualPort PPP
```

```
ADD !V1 -PPP AuthRemoteUser ("LS1", "LS1PW")
```



*At least one endpoint has to use SysCallID to authenticate, the other endpoint can choose to use either SysCallID or AuthRemoteUser.*

- 6 If SysCallerID is used in step 5, configure SysCallerID of this bridge/router using:

```
SETDefault -SYS SysCallerID="<srting>"
```

For example, if SysCallerID for this bridge/router is "NB", enter:

```
SETDefault -SYS SysCallerID="NB"
```

If AuthRemoteUser is used in step 4, configure local user names and passwords for the port that is going to connect to the peer using:

```
SETDefault !<port> -PPP AuthLocalUser = (["<userid>" | None],
"<password>")
```

For example, if the user name and password for PPP negotiation are "NB" and "NBPW", enter:

```
SETDefault !V1 -PPP AuthLocalUser = ("NB","NBPW")
```

- 7 Configure the parameters of the path from which the backup line will be dialed.

- 8 If the physical interface is a dial-up line, you need to configure your communication resources to use this line before issuing the Dial command. See the Configuring Port Bandwidth Management chapter in *Using Enterprise OS Software* for information about configuring communication resources for dial-up use.



You should not configure an access list in this configuration unless you know the range of the IP addresses that the ISP will assign to the remote peer.

- If L2TP is used as the tunneling protocol, set the L2TP local user name of the NETBuilder bridge/router using:

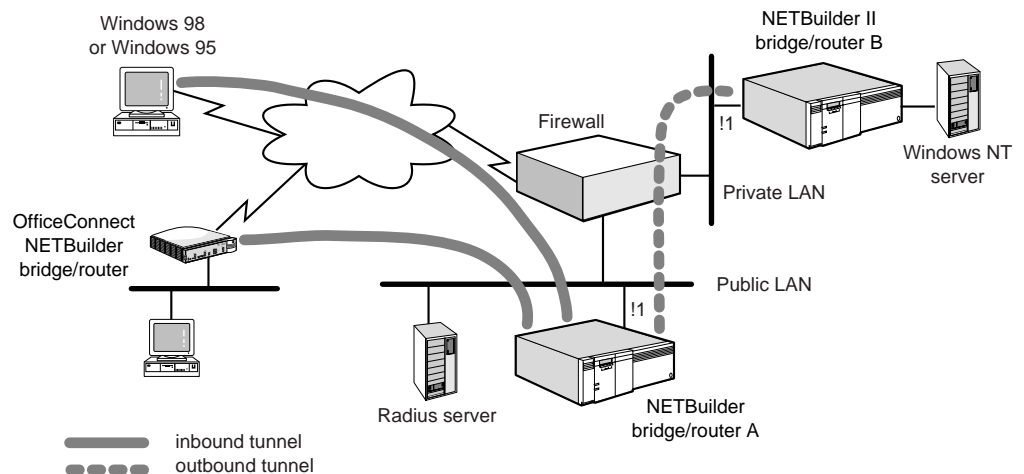
```
SETDefault -L2T L2TPLocalUser = "Name"
```

The assigned L2TPLocalUser name will be used as the "Host Name" during tunnel establishment which is required by the L2TP protocol.

## Configuring Tunnel Switching

To set up tunnel switching, private LAN tunnel users must first establish the tunnels (inbound tunnels) with the NETBuilder bridge/router. As Figure 85 shows, after the NETBuilder bridge/router authenticates the users, a second tunnel (outbound tunnel) is established between NETBuilder A and NETBuilder B. From then on, all the data packets from the remote users are switched to the NETBuilder B.

**Figure 90** Configuring Tunnel Switching



To configure NETBuilder bridge/router A to perform PPTP tunnel switching, follow these steps:

- Enable the L2TP Service. Enter:

```
SETDefault -l2t cont=e p=all
```

- Setup IP for physical port !1 on the NETBuilder A bridge/router. Enter:

```
SETDefault !1 -ip net=126.1.1.1 255.255.0.0
SETDefault -ip cont=ro
```

- Set up the TunnelSwitch on port !v1 on the NETBuilder bridge/router. Enter:

```
ADD !v1 -po VirtualPort TunnelSwitch
```

- Add DialNumberList to TunnelSwitch port.

The server address of DialNumberList must be the IP address of tunnel terminator. Assume the IP address of port !1 on NETBuilder bridge/router B is 126.1.2.2. Enter:

```
ADD !v1 -po DialNumberList"@126.1.2.2" type=pptp
```

- To enable L2TP tunnel switching, enter:

```
ADD !v1 -po DialNumberList"@126.1.2.2" type=L2TP
```

- 6 Add a remote user into AuthRemoteUser of TunnelSwitch port. Enter:

```
ADD !v1 -ppp aru "demo" "demopw"
SETDefault !v1 -po cont=e
```

### Configure Tunnel Switching Using the Radius Server

To configure tunnel switching using the Radius Server, follow these steps:

- 1 Enable L2T service:

```
SETDefault -l2t cont=e p=all
```

- 2 Set up IP for physical port !1 of NETBuilder A.

```
SETDefault !1 -ip net=126.1.1.1 255.255.0.0
SETDefault -ip cont=ro
```

- 3 Set up RAS service on NETBuilder A using the IP address 126.1.1.2 for the RADIUS server.

```
SETDefault -ras PrimAuthSrvr =126.1.1.2
SETDefault -ras PrimACntSrvr =126.1.1.2
SETDefault -ras Secret ="your secret"
SETDefault -ras SecurityType=RADIUS
SETDefault -ras cont=e
```

- 4 Follow the instructions that accompanied your RADIUS server to setup your RADIUS server. Enter:
- 5 Define the following tunneling attributes for the users performing tunnel switching:

- Tunnel\_Type: PPTP or L2TP
- Tunnel\_Medium\_Type: IP
- Tunnel\_Server\_Endpoint: 126.1.2.2 (IP address of the tunnel terminator)
- Tunnel\_Preference: 0

If the value of Tunnel\_Server\_Endpoint is not a local IP address of NETBuilder bridge/router A, NETBuilder bridge/routerA dynamically allocates a TunnelSwitch port.

---

## Tunnel Security

There are two levels of tunnel security implemented in L2TP: tunnel authentication and tunnel encryption.

Tunnel authentication (also known as tunnel challenge) is defined in the L2TP specification as an optional item.

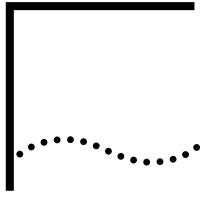
Tunnel authentication is similar to the CHAP in PPP and serves the same purpose. Tunnel authentication is operated at the L2TP protocol level (that is, at the box level), however, CHAP in PPP is operated at the PPP protocol level that is, at the end user level). The L2TPLocalUser and L2TPRemoteUser parameters are equivalent to *AuthLocalUser* and *AuthRemoteUser* in the PPP service, respectively.

L2TP tunnel encryption can be specified at the data packet level, control packet level or both. If the L2TPSecLevel parameter is specified as "None", no tunnel encryption is performed. If "Data" is specified, only data packets are encrypted. If



“Control” is assigned to the L2TPSecLevel parameter, only control packets are encrypted. If “Both” is specified, both data and control packets are encrypted.





# CONFIGURING TUNNEL ROUTE SHORT CUTS

This chapter describes how to configure a point-to-multi-point (P2MP) tunnel to discover a Tunnel Route Short Cut (TRSC) endpoint within a virtual private network (VPN). It also describes how to configure the Next Hop Routing Protocol (NHRP) to dynamically establish P2MP tunnels.

---

## Data Forwarding in a Tunnel

IP-Over-IP tunnels allow you to build VPNs across distant geographical internets. One of the most common topologies for a VPN is a central site network that uses VPN tunnels to connect to remote sites. This is particularly true in the case of a corporation that has a central office and remotely located branches.

Most often, the VPN is configured so that there is a statically configured tunnel between the central office network and each remote site. Using this configuration, in order for two remote sites to communicate, the data must travel an extra hop, through the central office network. As the VPN grows, this configuration can add many additional hops between the source and destination endpoints.

One way to make data forwarding more efficient would be to make the entire VPN fully meshed. In this case, each pair of routers would need to be explicitly defined. Obviously, creating a fully meshed VPN is a time-consuming task that is not easily scalable.

Using TRSC and NHRP, you can configure the remote sites to automatically discover each other. Once discovered, these sites can establish a dynamic virtual tunnel between them.

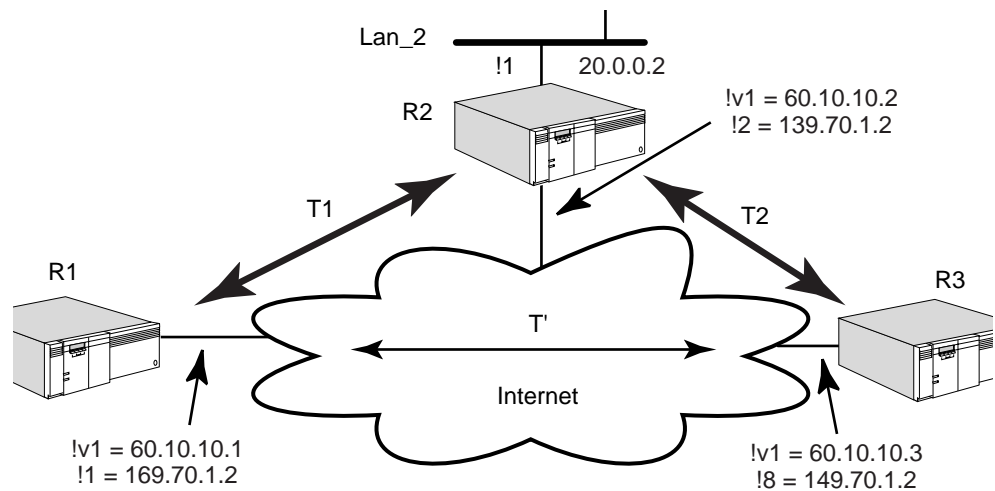
NHRP requires very little configuration and allows the routers to dynamically learn the best next hop node. By using the dynamic virtual tunnel, TRSC solves the problem of multiple hops and inefficient routed pathways.

---

## Configuring NHRP

NHRP allows router endpoints in a P2MP VPN to discover and resolve the best next hop tunnel end-node, thereby establishing a dynamic virtual tunnel on demand, in a partially meshed network.

Figure 91 is an example of a central site network that is connected in a star configuration to two remote offices.



**Figure 91** Central Site with Two Remote Site VPN

In Figure 91, R2 is the center site in the VPN. To establish a connected VPN without using NHRP, R2 requires two neighbor address configurations, one each for neighbor R1 and neighbor R3.

Using NHRP, the routing table and address table of each router is enhanced with NHRP learned short cut information to help make the routing path more efficient. Because the NHRP short cut is stored in the routing table, whenever the route is outgoing to an IPIP port, the router attempts to use its NHRP short cut instead of a statically configured path.

In this example, port 1 of R1 (169.70.1.2), port 2 of R2 (139.70.1.2), and port 8 of R3 (149.70.1.2) are physical paths that connect to the global Internet. Layered on top of these paths, an IP-over-IP VPN (using the address 60.10.10.xx) must be created that, in turn, connects the three LAN segments, Lan\_1, Lan\_2, and Lan\_3.

In the diagram, T1 and T2 are static IP-over-IP tunnels, manually configured by the system administrator. Without TRSC/NHRP, data traffic flowing from PC1 to PC3 must traverse router R2, because R2 is the only connected tunnel node. After TRSC and NHRP are enabled on the routers (R1, R2, and R3), traffic from PC1 to PC3 triggers the NHRP discovery mechanism and eventually establishes the dynamic virtual tunnel, T', between R1 and R3.

**Configuring R1** To configure router R1, follow these steps:

- 1 Assign a NETaddress to the physical path that connects to the global Internet.

For example:

```
SETDefault !1 -IP NETaddr=169.70.1.2
```

- 2 Enable IP routing.

For example:

```
SETDefault -IP CONTROL=ROute
```

- 3 Create a P2MP virtual port, V1.

For example:

```
ADD !V1 -Port VP IPIP P2MP
```

- 4 Assign a VPN NETaddress to V1.

For example:

```
SETDefault !V1 -IP NETaddr=60.10.10.1
```

- 5 Enable OSPF operation on port V1, so that R1 can exchange routing information with other connected VPN nodes.

For example:

```
SETDefault !V1 -OSPF CONTROL=Enable
```

- 6 Configure the neighboring tunnel endpoint address for VPN port V1.

For example:

```
ADD -IP ADDRESS 60.10.10.2 IPIP 139.70.1.2
```

- 7 Configure static and default routes on the VPN port and the physical port so that V1 is connected to the VPN and router R2 is connected to the global Internet.

For example:

```
ADD -IP ROUTe 0.0.0.0 169.70.1.1 3
```

```
ADD -IP ROUTe 60.10.10.3 60.10.10.2 5
```

- 8 Configure the private segment, Lan\_1. Assign an IP NETaddress and enable OSPF on port 3, so the routing information in the private segment is advertised into the VPN, and routing information from the VPN can be learned.

For example:

```
SETDefault !3 -IP NETaddr=10.0.0.2
```

```
SETDefault !3 -OSPF CONTROL=Enable
```

- 9 Enable TRSC to discover the short cut to reach other tunnel endpoints.

For example:

```
SETDefault !V1 -TRSC CONTROL=Enable
```

**Configuring R2** To configure router R2, follow these steps:

- 1 Assign a NETaddress to the physical path that connects to the global Internet.

For example:

```
SETDefault !2 -IP NETaddr=139.70.1.2
```

- 2 Enable IP routing.

For example:

```
SETDefault -IP CONTROL=ROute
```

- 3 Create a P2MP virtual port, V1.

For example:

```
ADD !V1 -PO VP IPIP P2MP
```

- 4 Assign a VPN NETaddress to V1.

For example:

```
SETDefault !V1 -IP NETaddr=60.10.10.2
```

- 5 Enable OSPF operation on port V1, so that R2 can exchange routing information with other connected VPN nodes.

For example:

```
SETDefault !V1 -OSPF CONTROL=Enable
```

- 6 Configure the neighboring tunnel endpoint address for VPN port V1.

For example:

```
ADD -IP ADDRESS 60.10.10.1 IPIP 169.70.1.2
ADD -IP ADDRESS 60.10.10.3 IPIP 149.70.1.2
```

- 7 Configure static and default routes on the VPN port and the physical port so that V1 is connected to the VPN and router R3 is connected to the global Internet.

For example:

```
ADD -IP ROUTE 0.0.0.0 139.70.1.1 3
```

- 8 Configure the private segment, Lan\_2. Assign an IP NETaddress and enable OSPF on port 1, so the routing information in the private segment is advertised into the VPN, and routing information from the VPN can be learned.

For example:

```
SETDefault !1 -IP NETAddr=20.0.0.2
SETDefault !1 -OSPF CONTROL=Enable
```

- 9 Enable TRSC to discover the short cut to reach other tunnel endpoints.

For example:

```
SETDefault !V1 -TRSC CONTROL=Enable
```

### Configuring R3 To configure router R3, follow these steps:

- 1 Assign a NETaddress to the physical path that connects to the global Internet.

For example:

```
SETDefault !8 -IP NETAddr=149.70.1.2
```

- 2 Enable IP routing.

For example:

```
SETDefault -IP CONTROL=ROUTE
```

- 3 Create a P2MP virtual port, V1.

For example:

```
ADD !V1 -PO VP IPIP P2MP
```

- 4 Assign a VPN NETaddress to V1.

For example:

```
SETDefault !V1 -IP NETAddr=60.10.10.3
```

- 5 Enable OSPF operation on port V1, so that R1 can exchange routing information with other connected VPN nodes.

For example:

```
SETDefault !V1 -OSPF CONTROL=Enable
```

- 6 Configure the neighboring tunnel endpoint address for VPN port V1.

For example:

```
ADD -IP ADDRESS 60.10.10.2 IPIP 139.70.1.2
```

- 7 Configure static and default routes on the VPN port and the physical port so that V1 is connected to the VPN and router R1 is connected to the global Internet.

For example:

```
ADD -IP ROUTe 0.0.0.0 149.70.1.1 3
ADD -IP ROUTe 60.10.10.1 60.10.10.2 5
```

- 8 Configure the private segment, Lan\_3. Assign an IP NETAddress and enable OSPF on port 4, so the routing information in the private segment is advertised into the VPN, and routing information from the VPN can be learned.

For example:

```
SETDefault !4 -IP NETaddr=30.0.0.2
SETDefault !4 -OSPF CONTROL=Enable
```

- 9 Enable TRSC to discover the short cut to reach other tunnel endpoints.

For example:

```
SETDefault !V1 -TRSC CONTROL=Enable
```

### Address and Routing Table Displays

Once configured, the address table of router R1 is as follows:

| Address    | Port | Media    | Address    | Owner       |
|------------|------|----------|------------|-------------|
| 60.10.10.1 | V1   | Local    |            | IPIP Static |
| 60.10.10.2 | V1   | External | 139.70.1.2 | IPIP Static |
| 60.10.10.3 | V1   | External | 149.70.1.2 | IPIP NHRP   |

The address table of router R2 is as follows:

| Address    | Port | Media    | Address    | Owner       |
|------------|------|----------|------------|-------------|
| 60.10.10.2 | V1   | Local    |            | IPIP Static |
| 60.10.10.1 | V1   | External | 169.70.1.2 | IPIP Static |
| 60.10.10.3 | V1   | External | 149.70.1.2 | IPIP Static |

The address table of router R3 is as follows:

| Address    | Port | Media    | Address    | Owner       |
|------------|------|----------|------------|-------------|
| 60.10.10.3 | V1   | Local    |            | IPIP Static |
| 60.10.10.2 | V1   | External | 139.70.1.2 | IPIP Static |
| 60.10.10.1 | V1   | External | 169.70.1.2 | IPIP NHRP   |

The route entries on router R1 with regard to PC1, PC2, and PC3 are as follows:

| Destination | Mask      | Gateway    | Source    |
|-------------|-----------|------------|-----------|
| 10.0.0.0    | 255.0.0.0 | 10.0.0.2   | Connected |
| 20.0.0.0    | 255.0.0.0 | 20.0.0.2   | OSPF      |
| 30.0.0.0    | 255.0.0.0 | 60.10.10.3 | OSPF      |

The routing entry on router R2 with regard to PC1, PC2, and PC3 is as follows:

| Destination | Mask      | Gateway    | Source    |
|-------------|-----------|------------|-----------|
| 10.0.0.0    | 255.0.0.0 | 60.10.10.1 | OSPF      |
| 20.0.0.0    | 255.0.0.0 | 20.0.0.2   | Connected |
| 30.0.0.0    | 255.0.0.0 | 60.10.10.3 | OSPF      |

The routing entries on router R3 with regard to PC1, PC2, and PC3 are as follows:

| Destination | Mask | Gateway | Source |
|-------------|------|---------|--------|
|-------------|------|---------|--------|

|          |           |            |           |
|----------|-----------|------------|-----------|
| 10.0.0.0 | 255.0.0.0 | 60.10.10.3 | RIP       |
| 20.0.0.0 | 255.0.0.0 | 60.10.10.2 | RIP       |
| 30.0.0.0 | 255.0.0.0 | 60.10.10.3 | Connected |

## Overview of NHRP

The TRSC Service uses NHRP to allow a tunnel endpoint to resolve the best next hop endpoint so that a dynamic tunnel can be established in a P2MP VPN. Before it can establish a TRSC, each endpoint router must terminate at least one statically configured tunnel (the default path), and all participating nodes must be connected.

Each NETBuilder router in the VPN works as both an NHRP client and an NHRP server.

### Router Kicks Off an NHRP Resolution Request

When routing a packet, the routing protocol first looks up the next hop for a destination. If the routing protocol determines the next hop gateway is via an IPIP port (the VPN), the router attempts to retrieve the NHRP short cut. If a short cut is not available, the router forwards the packet to the default (static) tunnel path towards the destination.

In addition to forwarding the packet, the router kicks off an NHRP Resolution Request for the destination using the VPN path. The router issues an NHRP Resolution Request when the following are true:

- There is a data packet to be routed.
- The target destination is located on the VPN network, via an IPIP port route entry.
- NHRP short cut information for the route is not available in the route table.

### Router Receives an NHRP Resolution Reply

When a router receives a valid NHRP Resolution Reply, the router saves the NHRP information in its ClientCache table. It adds the learned NHRP information to its local routing and address tables, using NHRP as the owner type.

### Router Receives an NHRP Resolution Request

When the router receives an NHRP Resolution Request, it does one of three things:

- Drops the packet.

The router first attempts to look up a route for the target destination. If a route does not exist, the router drops the packet without further action.

- Forwards the request.

If the route exists, and it is toward an IPIP port, and the TRSC Service is enabled on that outbound IPIP port, the router attaches its own address to the packet and forwards it.

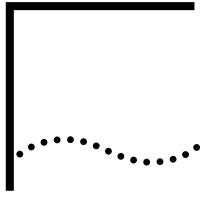
- Replies to the request using an NHRP Request Reply.

The router determines that it is the next hop server (NHS) because the destination network is located via a non-IPIP port or a port that does not have TRSC enabled. The router (NHS) replies to the Resolution Request using an NHRP Resolution Reply along the reverse path from which the Resolution Request was received.



The NHS stores the source <network layer address> and <media layer address> pair in its address table. The NHS also makes an entry in its ServerCache table, and tracks the ValidTime and HoldTime.





# CONFIGURING IPv6 ROUTING

This chapter describes the procedures for configuring your system to perform Internet Protocol Version 6 (IPv6) routing. It describes how the router works and provides guidelines for operating, managing, and troubleshooting it.



*For conceptual information, see “How the IPv6 Router Works” later in this chapter.*

---

## Configuring a Basic IPv6 Router

The procedure in this section describes the minimum number of steps required to configure your system to route IPv6 packets. Depending on your network requirements, you can use the default values of the parameters in the various services, or you can further configure the router according to later sections in this chapter.

To configure the IPv6 router, you must set parameters in the Routing Information Protocol Next Generation (RIPNG) Service if your network uses the RIPNG as the interior gateway protocol for IPv6 intra-autonomous system routing.

You can set parameters in the BGP service if your network uses BGP as the exterior gateway protocol for IPv6 inter-autonomous system routing.

The IPv6 parameters enable the IPv6 routing function and configure the networks connected to the router. The following information describes how to configure IPv6 routing.

## Configuring for Local Area Networks

Use this procedure to configure basic IPv6 routing over LAN (Ethernet/FDDI) ports and Point-to-Point Protocol (PPP) links.

### Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your router and log on to the system with Network Manager privilege.
- Become familiar with the protocols supported by the router.
- Obtain an IPv6 address for each port you want to configure.

### Procedure

To set up a basic configuration for your IPv6 router, follow these steps:

- 1 Assign an IPv6 address for each port that will perform IPv6 routing using:

```
ADD !<port> -IPV6 NETaddr <IPv6 address>
```

The IPv6 address format and its textual representation must be consistent with the Internet draft on IPv6 Addressing Architecture, version 2.

The local interface identifier is generated using the EUI-64 format. For backward compatibility, static routes and dynamically learned routes that conform to RFC 1884 are also supported.

- 2 Enable the dynamic routing protocols (RIPNG) for IPv6 routing.

To enable RIPNG operation on a specified port, set the CONTROL parameter in the RIPNG Service (using its TALK and Listen values) as follows:

```
SETDefault !<port> -RIPNG CONTROL = ([TALK | NoTALK],
 [Listen | NoListen], [Poison | NoPoison], [TRigger | NoTRigger])
```

Setting the CONTROL parameter to the TALK and Listen values enables the router to send and receive routing information with other routers using RIPNG.

- 3 Enable IPv6 routing on a specified port by entering:

```
SETDefault !<port> -IPV6 CONTROL = ROute
```

### Related Information

By default, a link-local address is assigned to an IPv6 routing interface. For a transit router with no attached hosts, the interface-specific subnet prefix assignment may be omitted. In this case, the router can communicate with other nodes on the same link by using the interface's link-local address as its source address.

### Configuring for Wide Area Network Connectivity

IPv6 routing over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), X.25, and ATM Emulated LAN is currently not supported.

WAN connectivity to remote IPv6 networks is supported via PPP links, ATM PVCs, and tunneling IPv6 over IPv4 configured tunnels.

IPv6 routing over ATM PVCs is supported over a fully meshed topology only. To run RIPNG over ATM PVC, the list of neighborhood routers must be configured by using the RIPNG AdvToNeighbor parameter.

---

### Verifying the Configuration

To verify the configuration, examine network devices and send packets from one network to another to determine if they are properly forwarded.

### Examining Network Devices

To examine the status of the IPv6 router, follow these steps:

- 1 Display information on the attached networks by entering:  
**SHoW -IPV6 NETaddr**
- 2 Determine which stations or networks are reachable from the router by entering:  
**SHoW -IPV6 AllRoutes**
- 3 Display information from the Address Translation Table by entering:  
**SHoW -IPV6 ADDRess**
- 4 Display information on the configured tunnels by entering:  
**SHoW -IPV6 TUnnel**
- 5 Examine the neighbor discovery status by entering:  
**SHoW -IPV6 NbrDiscovery**

**Getting Statistics** To examine the IPv6 statistics, follow these steps:

- 1 Examine the IPv6 statistics by entering:

```
SHoW -SYS STATistics -IPv6
```

- 2 Examine the RIPNG statistics by entering:

```
SHoW -SYS statistics -RIPNG
```

- 3 Examine the BGP statistics. Enter:

```
SHoW -sys statistics -BGP
```

You can collect statistics for a specific period by using the SampleTime and STATistics parameters. For more information, see *Reference for NETBuilder Family Software*.

**Checking the Overall Status** To check the overall status of the IPv6 router, follow these steps:

- 1 Examine the path configurations by entering:

```
SHoW -PATH CONFIguration
```

- 2 Examine the port configurations by entering:

```
SHoW -PORT CONFIguration
```

- 3 Examine the IPv6 configurations by entering:

```
SHoW -IPv6 CONFIguration
```

- 4 Examine the RIPNG configurations by entering:

```
SHoW -RIPNG CONFIguration
```

- 5 Examine the BGP configuration. Enter:

```
SHoW -BGP configuration
```

---

### Checking with Ping6, TraceRoute6, and Telnet6

After you have configured the router for IPv6, check to determine if the router can forward packets properly by following these steps:

- 1 Use the Ping6 command to verify proper routing to each of the other routers or hosts.
- 2 If the Ping6 fails, use the TraceRoute6 command to determine where the route fails.
- 3 Use the Telnet6 command to access the last router that responds to the TraceRoute6 command. The configuration, IPv6 routing table, and status of the router can be examined and the basic tools (Ping6, TraceRoute6, and Telnet6) can be used to troubleshoot the problem.

Ping 6 TraceRoute 6 and Telnet 6 commands work exactly like the IPv4 counterparts. However, if the target address is not a link-local address, an IPv6 address with the same or higher scope (site or global), must be assigned to either the internal port or the port through which the target address can be reached.

## Customizing the IPv6 Router

After you set up and check the configuration of the basic IPv6 router, it is ready to perform packet routing. If desired, you can further customize your IPv6 router by doing the following tasks:

- Configure the internal port
- Configure multiple subnets
- Configure neighbor discovery
- Configure static routes
- Configure RIPNG routing policies
- Configure RIPNG route aggregation
- Configure IPv6 routing over ATM using PVCs
- Configuring Interautonomous System routing using the BGP multiprotocol extensions

## Configuring the Internal Port

The IPv6 link-local subnet exists by default once IPv6 routing is enabled on an interface, therefore, it is not necessary to configure a IPv6 subnet on an interface to perform IPv6 routing. Since a local address is required to support end-to-end communication with remote nodes for applications such as Ping6 and TELNET 6, a node address must be assigned to the router by configuring an internal port. To configure the internal port, follow these steps:

- 1 Configure a single subnet prefix on the internal port:

```
ADD !0 -IPV6 NETaddress 3ffe:1:1:1/64
```

The MAC address of the CPU board is used to generate the EUI-64 interface identifier and append to the configured prefix to form the global IPv6 address.

- 2 Activate the internal port by entering:

```
SETD !0 -IPV6 CONTrol=ROute
```

The internal port stays up if IPv6 is active on one or more interfaces. The address assigned to the internal port is used as a local address for end system packets originated from the router. The internal port address is advertised by the routing protocol as a reachable subnet prefix.

## Configuring Multiple IPv6 Subnets

Your IP router supports multiple IPv6 subnets. You can configure more than one IPv6 network or subnet on a port. The procedure for configuring multiple IPv6 subnets on all interfaces is the same. The following example configures multiple IPv6 subnets on an Ethernet network.

This example is a topology where two IPv6 networks are configured on an Ethernet. Use this example to configure these IPv6 networks on Ethernet.

- 1 Enter:

```
ADD !<port> -IPV6 NETAddr <IPv6 address>
```

- 1 To delete a subnet, enter:

```
DELeTe !<port> -IPV6 NETAddr <IPv6 address>
```

## Configuring Neighbor Discovery

The router supports IPv6 Neighbor Discovery only on LAN ports according to RFC 1970. To configure neighbor discovery, use:

```
SETD !<port> NbrDiscovery = ([DupAddrDetect | NoDupAddrDetect],[
 SendRouterAdv | NoSendRouterAdv],[StatefulAddrConf |
 NoStatefulAddrConf],[OtherStatefulConf | NoOtherStatefulConf],[
 AdvHopLimit | NoAdvHopLimit], [AdvMTU | NoAdvMTU] [AdvReachableTime |
 NoAdvReachableTime],[AdvRetryTime | NoAdvRetryTime],
```

Link layer address resolution is enabled by default.

Duplicate address detection can be enabled with the NbrDiscovery parameter. If the interface MAC address already exists when the link comes up, routing will be disabled on the interface and a system message is logged.

Autoconfiguration of adjacent hosts can also be configured with the NbrDiscovery parameter. In conjunction with the NetAddress parameter options, the neighbor hosts acquire the configured subnet prefix and link-specific network parameters advertised by the router.

## Configuring Static Routes

A static route is a user-defined route by which a network can be reached. You can configure as many static routes as desired.

### Procedure

To add a static route, use:

```
ADD !<port> -IPV6 ROUTe <IPV6 address> [<gateway>] <metric> [Override]
```

To delete a static route, use:

```
DELete -IPV6 ROUTe <IPV6 address> [<gateway>]
```

This example shows that the subnet 3ffe:1:1:3/64 is reachable on port 2 through the gateway fe80:200:81ff: fed5:0892. The metric is 3.

For example, enter:

```
ADD !2 -IPV6 route 3ffe:1:1:3/64 fe80:200:81ff:fed5:0892 3
```



*The gateway must be located on a network directly connected to the router on which you add the static route.*

If the outgoing interface is a point-to-point configured IPv4 tunnel, you can add a static route without the next-hop gateway address. For example:

```
ADD !t1 -IPV6 Route 3ffe:1:1:5/64 2
```

## Configuring RIPNG Routing Policies

The routing policies supported by RIPNG allow you to control the reporting of routing information on a per-port basis. This section describes the various routing policies you can configure and the parameters associated with configuring each policy, and provides examples of configuring policies.

### Prerequisites

Before beginning this procedure, complete the following tasks:

- Familiarize yourself with the various policies that are available, then determine which policies you want to configure. Table 22 lists and briefly describes each policy and its associated parameter.

- If you plan to access or receive information on routes from specific networks as opposed to all or no networks, determine the IPv6 addresses of these specific networks.

**Table 22** RIPNG Routing Policies

| Policy    | Description                                                                | Parameter       |
|-----------|----------------------------------------------------------------------------|-----------------|
| Advertise | Controls which routes are reported regardless of the route source.         | AdvertisePolicy |
| Static    | Controls which static routes are reported in the IPv6 routing environment. | StaticPolicy    |
| Exterior  | Controls which BGP routes are reported                                     | ExteriorPolicy  |
| Receive   | Controls which RIPNG routes are accepted by a trusted neighbor.            | ReceivePolicy   |

For more information on the parameters listed in this table, see the RIPNG Service Parameters chapter in *Reference for NETBuilder Family Software*.

### Procedure

To configure a routing policy, follow these steps:

- 1 Establish an advertise policy that controls the advertisement of routes through RIPNG regardless of the source from which the route is learned. Use:

```
ADD !<port> -RIPNG AdvertisePolicy All | None | [~]<IPv6 address>
 [<metric> (0-15)]
```

For example, to configure a policy on port 1 that forwards information on all routes to network 3FFE:1:1:1/64, enter:

```
ADD !1 -RIPNG AdvertisePolicy 3FFE:1:1:1/64
```

In this example, a metric associated with network 3FFE:1:1:1/64 was not specified. If you decide not to specify a metric with the AdvertisePolicy parameter or to specify a metric of zero, a route is reported with a metric calculated from the routing table.

- 2 Establish a receive policy that accepts or rejects routes learned by RIPNG from trusted neighbors. Use:

```
ADD !<port> -RIPNG ReceivePolicy All | None | [~]<IPv6 address> [<metric>
 (0-15)]
```

For example, to configure port 1 so that it accepts information on routes learned by RIPNG for network 3FFE:1:1:1/64, enter:

```
ADD !1 -RIPNG ReceivePolicy 3FFE:1:1:1/64
```

In this example, a metric associated with network 3FFE:1:1:1/64 was not specified. If you decide not to specify a metric with the ReceivePolicy parameter or specify a metric of zero, a route with the originally reported metric is stored in the routing table.

- 3 Establish a static policy for reporting static routes.

For example, to configure a policy on port 1 that forwards routing information about static routes configured on network 3FFE:1:1:1/64, enter:

```
ADD !1 -RIPNG StaticPolicy 3FFE:1:1:1/64
```



In this example, a metric associated with network 3FFE:1:1:1/64 was not specified. If you decide not to specify a metric with the StaticPolicy parameter or specify a metric of zero, a route is reported with a metric calculated from the routing table.

- 4 Establish an exterior policy for reporting routes learned by BGP. For example, to configure a policy on port 1 that advertises all the BGP routes except 3ffd/64, enter:

```
ADD !1-RIPNG ExteriorPolicy 3ffd/64
```

### Configure IPv6 Routing Over ATM using PVCs

To configure IPv6 routing over ATM using PVCs, follow these steps.

- 1 Set up the ATM Service. For example, enter:

```
ADD !V1 -port VirtualPort 1 MPATM
ADD !V1 -ATM PVC 11 60.11
```

- 2 Obtain the IPv6 link-local address and the local VCID of the PVC for each neighbor router that is attached to the ATM switch and has participated in the fully meshed topology.
- 3 Assuming a transit router in the ATM domain, the link-local address is used to communicate with peer neighbor routers. IPv6 subnet assignment to an ATM port is not required.
- 4 Specify IPv6 link-local address to ATM PVC mapping for all neighbor routers.
- 5 Map the neighbor router's link-local address to the local PVC associated with the neighbor router. For example, enter:

```
ADD !V1 -IPV6 Address fe80::1 &11
```

11 is the local VCID of the PVC established between the local router and the neighbor router with the link-local address fe80::1. "ATM" can be used in place of the & sign.

- 6 Specify the list of neighbor routers to exchange routing information through RIPNG using the AdvToNeighbor parameter. For example, enter:

```
ADD !V1 -RIPNG AdvToNeighbor fe80::1
```

- 7 Enable IPv6 routing and RIPNG on the ATM virtual port, by entering.

```
SETD !V1 -IPV6 CONTrol = ROUTe
SETD !V1 -RIPNG CONTrol = (TAlk, LISten)
```

- 8 Perform steps 1-6 on all peer neighbor routers.

### Configuring RIPNG Route Aggregation

Route aggregation allows multiple specific routes to be advertised as a single prefix route. By combining several network/host routes into one subnet route, the router update message is smaller and the size of the routing table is reduced.

To configure RIPNG route aggregation, follow these steps:

Specify a prefix route that RIPNG advertises as a simple supernet route on a port using:

```
ADD !<port> -RIPNG AggregateRoute <IPv6 prefix> [metric]
```

When RIPNG route advertisement is enabled with SETD !<port> -RIPNG CONTrol=TA, all configured aggregate routes are advertised while all the associated sub-routes advertisements are suppressed.

## How the IPv6 Router Works

This section describes the following concepts involved in IPv6 routing:

- Understanding IPv6 network topology
- Multipath routing
- Default routes
- Learning routes within an autonomous system
- Transition Support
- Interautonomous system routing via BGP multiprotocol extensions.

## Understanding IPv6 Network Topology

An IPv6 network is configured on each interface where IPv6 packets are received and sent. The interface can be either a local Ethernet or FDDI interface, a PPP link or a Multi protocol over ATM (MPATM) virtual port.



**CAUTION:** Each IPv6 subnet that you assign directly to a port must be unique, that is, you cannot assign the same IPv6 subnet to different ports.



Only RIPNG is supported for interautonomous system routing.

A router must check its routing table to determine where to route a packet. If the destination is on an attached network, the router can send it directly to the network. But if the destination is farther away in the internetwork, the router must route the packet to another router (called a *gateway*) that is closer to the destination. The route to a gateway can be statically configured or dynamically learned through RIPNG. When two routers are located on the same network (that is, each of them has at least one interface to the network), they are considered *neighbors* or *neighboring gateways*.

## Multipath Routing

The router supports multipath routing, which means that up to four routes for each destination address can be stored in the routing table. Advantages of multipath routing are as follows:

- The router can still route a packet using an alternative route if the primary one fails; it is more responsive to network topology changes than if only one route to a destination exists.
- The router can distribute the load among the available equal-cost best paths.

When multiple routes for a destination exist, the router uses the route with the highest precedence. The types of routes used (listed by decreasing precedence) are listed in Table 23.

**Table 23** Route Precedence

| Precedence Level | All Protocols without BGP       | With BGP                        |
|------------------|---------------------------------|---------------------------------|
| 1                | Static (added without Override) | Static (added without override) |
| 2                | RIPNG                           | RIPNG                           |
| 3                | ICMP redirect (host only)       | ICMP redirect (host only)       |
| 4                | Static (with override)          | Static (with override)          |
| 5                |                                 | BGP                             |
| 6                | Default route                   | BGP default route               |

The routing table displays routes with a high precedence first.

If the route with the highest precedence fails, the route with the next highest precedence will be used. A route in the routing table is deleted in these situations:

- A dynamic route learned through RIPNG is deleted when a router times out and goes through the HOLD-DOWN and GARBAGE COLLECTION states. A router times out when it fails to hear from a neighbor for a period that is six times the value of the UpdateTime parameter. For example, if the value of the UpdateTime parameter is 45 seconds, the router will time out if it does not hear from its neighbor for 270 seconds.
- A DELeTe ROUte command removes a static route.
- A lowest precedence route is deleted when four routes of higher precedence are available. This situation occurs when a fifth route is learned and has a higher precedence than the lowest precedence route.

Dynamic routes learned by RIPNG can be removed by using the FLush -IPV6 AllRoutes command.

### Route Selection and Load Splitting

If two or more routes with the same route source precedence are available to reach a destination, the router always selects the route with the lowest metric (measured in hops for RIPNG). If there is more than one route learned by the same routing protocol with the same equal-cost, low metric, you can split the load between these routes on a round-robin basis. The SplitLoad parameter (Enable | Disable) determines whether load splitting is performed.

Because load splitting balances the load among different routes, 3Com recommends it if two or more routes are available to reach a destination and the routes have similar metrics. However, if the routes connecting various networks have different metrics (that is, there is only one route with the fewest hops or lowest cost to a destination), load splitting is not necessary.

### Default Routes

When a router needs to route a packet destined for an address for which there are no entries in the routing table, it uses the default route if one exists. The network `::/0` represents the default route.

The router supports up to four default routes; when more than one default route is available, the same selection rules apply. If load splitting is enabled, the load is distributed among equal-cost best paths.

An advantage of a router using a default route is that network overhead in an autonomous system can be reduced. The reduction in overhead occurs because the router does not need to advertise all external routes.

The following example will help you understand default routes.

*Example* Router A receives a RIPNG update packet from router B, which has an entry indicating that network `::/0` is reachable with metric 3. Router A considers router B its default gateway. That is, if router A needs to route a packet whose destination is not found in its routing table, it sends the packet to router B.

To configure RIPNG to advertise a default route, assign a non-zero value to the `DefaultMetric` parameter. You do not need to configure the `DefaultMetric` parameter on every router throughout the domain. The default route learned on one interface is propagated to neighbors on the other interfaces.

RIPNG propagates the advertisement of the default route as the normal operation. For the RIPNG Protocol, it is possible to suppress propagation of the default route by using the `AdvertisePolicy` parameter.

You can configure the default route in one of two ways:

- On the exit router of a domain, configure a static override default route with a metric of 1 that points to the first hop outside the domain. Then use the `StaticPolicy` parameter of RIPNG to import this route into the routing protocol and advertise it into the domain.
- At the top level, set the `DefaultMetric` parameter in RIPNG to instruct the router to originate a default route.

### Learning Routes with RIPNG

The router supports RIPNG according to RFC 2080. Normally, every 30 seconds or every time it learns a route change for a network, the router uses multicast packets to report to its neighboring gateways the following types of information:

- The networks it can reach
- The metric associated with each network it can reach

By default, the information in update packets pertains only to learned routes. Static route information is not reported.

You can configure some router parameters to determine how the router sends out the updates and what is included in them. For example, you can configure the parameters for the following purposes:

- To change the frequency of the multicast traffic (`UpdateTime` parameter)
- To prevent the router from sending or receiving update and request packets (`CONTROL` parameter)
- To control the set of neighboring routers from which the router receives updates and to which it sends them (`AdvToNeighbor` and `RcvFromNeighbor` parameters)
- To prevent the router from sending out a trigger update response upon a route change for a network (`CONTROL` parameter)
- To enable the router to report static routes (`StaticPolicy` parameter)
- To cause some routes not to be reported or to be reported with the infinity metric, that is, using poison reverse (`CONTROL` parameter)

### Network Reachability

The following types of networks are considered *reachable* when a router multicasts its RIPNG update packets:

- All directly connected networks
- All static routes (as controlled by the `StaticPolicy` parameter)
- All dynamic routes learned through RIPNG

### Solving the Slow Convergence Problem with Split Horizon

When the router advertises its own reachable networks, it advertises to all neighbors except the one from which it learned of the network being advertised.

On a LAN network, it is not necessary to configure neighbors. If you do not configure neighbors, RIPNG multicasts the updates over the LAN. If you configure neighbors, RIPNG unicasts the updates to each of the neighbors on the AdvToNeighbor list

### Solving the Slow Convergence Problem with Poison Reverse

Poison reverse is configurable using the Poison or NoPoison value for the -RIPNG CONTROL parameter.

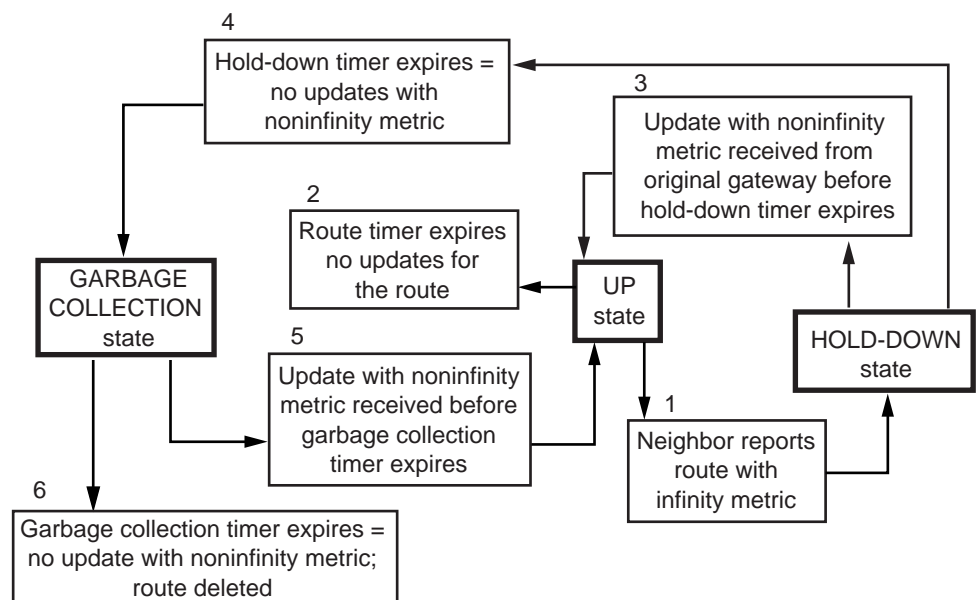
If poison reverse is enabled, the router advertises all routes to all neighbors, but when advertising a route to a neighbor that has advertised the same route, the router sets the metric to infinity (0xFFFF) to prevent the recipient from adding the route to its routing table. Poison reverse speeds convergence but adds to network overhead.

If poison reverse is disabled, the router omits routes learned from one neighbor from RIPNG updates sent to that neighbor. No poison reverse has the advantage of minimizing network overhead in large network configurations at the expense of slower convergence.

### Different States of RIPNG-Learned Routes

To avoid routing loops, new information about a route is ignored for a designated period before it is used. Figure 92 summarizes how a route learned through RIPNG changes states. Explanations of the different states follow the figure.

**Figure 92** Different States of a RIPNG Route



- **GARBAGE COLLECTION state**

When the timer for a route that has been in the HOLD-DOWN state expires, that route changes to GARBAGE COLLECTION state. This happens when no

update packets are received to indicate that the route is still reachable. In this state, if a neighboring gateway reports the route with a noninfinity metric within 120 seconds, the route can go back to the UP state. If no updates are received within 120 seconds (garbage-collection timer), the route is deleted from the routing table. It is possible to go into GARBAGE COLLECTION state if no updates are received within 180 seconds.

- UP state

A route is considered UP if it is reachable with a noninfinity metric (15 or fewer hops). Whether it is reachable is determined by the last update received from the neighboring gateways. It remains UP for 180 seconds (the route timer). The timer is reset each time a new update for the route is received.

- HOLD-DOWN state

A route in UP state changes to HOLD-DOWN state if an update received from the original gateway indicates that the route is associated with an infinity metric (16 hops). In this state, all update information received from other gateways for that route is ignored.

However, if an update is received from the original gateway within 60 seconds (the hold-down timer), and it associates a noninfinity metric with the route, the route goes back to UP state.

If the hold-down timer expires, the route goes from HOLD-DOWN state to GARBAGE COLLECTION state for 120 seconds.

When you display the routing table with the `SHoW -IPV6 AllRoutes` command, the state of each route is displayed under the `STATUS` heading.

## IPv6 Transition Support

The router support the following IPv6 transition mechanisms to provide IPv6 connectivity with remote dual stack (IPv4/IPv6) hosts/routers over the existing IPv4 routing infrastructures.

### Automatic Tunnel

Automatic tunnel does not require the configuration of tunnel end-points. It is used to communicate with a remote node (via Telnet6/Ping6/) using the IPv4-compatible address. The router can originate end system packets destined to a remote IPv4-compatible address and only receive end system packets destined to a local IPv4-compatible address. Otherwise, all packets received with a IPv4-compatible destination address will not be forwarded into the IPv6 routing domain and are discarded.

Enable automatic tunnel to enable by entering:

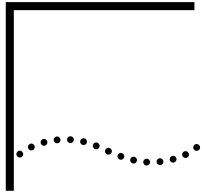
```
SETD !t0 -IPV6 CONTrol=ROUte
```

### Point-to-Point Tunnel

A configured tunnel requires the configuration of two tunnel end-points. It is treated as a virtual interface. IPv6 subnet prefixes can be assigned and IPv6 static routes can be configured to specify routes reachable via the configured tunnel. RIPNG can also be enabled on the configured tunnel.

To configure the tunnel, enter:

```
SETD !<tunnel id> -IPV6 tunnel=<local IPV4 addr> <remote IPV4 addr>
SETD !<tunnel id> -IPV6 CONTrol=ROUte
```



# CONFIGURING NETWORK ADDRESS TRANSLATION

This chapter describes the Network Address Translation (NAT) feature and how to configure network addresses for translation. This feature allows the IP addresses of a network to be translated into addresses that can be used outside the network. Do not use IPSEC and NAT together.



*For conceptual information, see “How NAT Works” later in this chapter.*

---

## Configuring Network Address Translation

The procedures in this section describe how to configure NAT.

### Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router using the procedure in the Configuring Basic Ports and Paths chapter in *Using Enterprise OS Software*.

### Enabling NAT Ports

To enable NAT on a port connected to an outside network, use:

```
SETDefault !<port> -NAT CONTROL = Enable
```

### Defining the Address Mapping

Address mapping allows you to map one set of IP addresses to another.

To configure the address map, use:

```
ADD !<port> -NAT AddressMap <LHS Address(es)> <RHS Address(es)> [InBound | OutBound | BiDirectional | LoadShare] [Log [0..7]] (<LHS and RHS Address(es)> = <IPaddr/mask> | <IPaddr> | <IPaddr>-<IPaddr> [, <IPaddr/mask> | <IPaddr> | <IPaddr>-<IPaddr>..])
```

Where LHS Address(es) and RHS Address(es) is a single address or a range of addresses.



*If you specify BiDirectional, you can use only a one-to-one map.*

Address maps take effect immediately. When you remove a map using the DELETE command, all session table entries based on the map are deleted.

If you have static and dynamic maps, the bridge/router checks the static map first and then the dynamic map. See “Address Mapping” later in this chapter for more information about static and dynamic maps.



*You can also enter address maps directly into the natmap file. The natmap file in your configuration file directory is in ASCII text format that can be edited with any text editor.*

If you choose to manually edit the `natmap` file, you must enter the `NATReStart` command or reboot the bridge/router to load the maps listed in the file.

The `NATReStart` command examines the `natmap` file for syntax errors. If a syntax error is encountered in the `natmap` file, processing stops, an error message is displayed, and no further maps are initialized.

*One-to-one example* The following command establishes a one-to-one map for BiDirectional connections on port 1. The LHS address can be inside or outside.

```
ADD !1 -NAT AddressMap 144.195.48.20 144.195.40.17 BiDirectional
```

*Many-to-one example* The following command establishes a many-to-one map for OutBound connections on port 1. The LHS addresses are inside, and the RHS address is outside.

```
ADD !1 -NAT AddressMap 192.168.0.0/16 144.195.18.4 OutBound
```

*Many-to-many example* The following command establishes many-to-many map for OutBound connections on port 1. The command maps four inside addresses to 16 outside addresses.

```
ADD !1 -NAT AddressMap 192.168.0.1-192.168.0.3, 192.168.10.17
144.195.40.0/28 OutBound
```

*One-to-many example* The following command establishes a one-to-many map for LoadShare connections on port 1. The LHS address is outside, and the RHS addresses are inside.

```
ADD !1 -NAT AddressMap 144.195.18.4 192.168.0.0/16 LoadShare
```

## Defining TCP/UDP Port Mapping

To map an address and TCP/UDP port to another address and TCP/UDP port, use:

```
ADD !<port> -NAT TcpUdpPortMap <IPaddr>, <TCP/UDP port#> <IPaddr>
[,<TCP/UDP port#>] [InBound | Outbound | BiDirectional] [Log[0..7]]
```

If you do not specify the second TCP/UDP port number, the software uses the same port number specified on the first address.

For example, to allow inbound telnet traffic to host 10.0.0.1 on port 1, enter:

```
ADD !1 -NAT TcpUdpPortMap 144.195.48.20,23 10.0.0.1 Inbound
```



*The software cannot differentiate between TCP and UDP ports.*

If you have address maps and TCP/UDP maps defined, the bridge/router checks the TCP/UDP map first and then the address maps.

## Logging Messages

The AuditLog Service controls the delivery of messages to the syslog server(s). Only the start-of-connection packets are logged, to avoid flooding logging messages. No more than 10 log messages per second are generated. All messages over 10 are suppressed and the next log message (generated after the one second window expires) contains a counter of how many previous messages were suppressed.

To log messages to the AuditLog Service, enable the service using:

```
SETDefault -AuditLog CONTrol = (COnfig, MESSAGES, Scurity)
```



To specify whether messages are logged to the AuditLog Service, the local console, or both, use:

```
SETDefault -NAT Log = [Syslog | NoSyslog] [Console | NoConsole]
[SessionFail | NoSessionFail] [SessionSuccess | NoSessionSuccess]
[LogDetail | NoLogDetail]
```

### Session Information

You can display the address translations, usage, and idle time statistics for active NAT sessions. Each translation is identified by a session ID. When no option is specified, all active NAT sessions are displayed. The FTP session display can have two TCP connections active at the same time. One is the control channel, for passing commands and responses between client and server. The other is the data channel for the actual data transfer.

To display the NAT session information, use:

```
SHow [!<port> | !*] -NAT SESSions [TCP | UDP | FTP | Others]
```

To determine the maximum time-out period allowed for NAT sessions to remain idle before the session is terminated, use:

```
SETDefault !<port> -NAT SessionTimeout [TCP | Others] <minutes>
(0-99999) [:<seconds>(0-59)]
```

### Translation failure actions

To specify how to handle a packet if the address translation fails at the start of a session, use:

```
SETDefault !<port> -NAT XlateFailAction = PassThrough | Drop |
GenerateICMP
```

Translation failure can occur in the following cases:

- The bridge/router ran out of mappable IP addresses, TCP/UDP port resources, or other internal resources such as memory.
- The session direction was not permitted, even though there was a match for the address in the map.

### Adding a Dynamic Address Map

To set up a dynamic address map for IP addresses assigned per session, use:

```
SetD !<portlist> -NAT IPCPAddressMap = < Enable > [LHS Address(es)] [Log
{0...7}]
```

When this address map is enabled, the port control must be toggled. Default address is 0.0.0.0./0 if not specified otherwise.

---

## How NAT Works

This section uses the following terms:

- *Inside network* — the network that includes addresses you want to map.
- *Outside network* — the network, such as the Internet, that you want to connect the inside network to.
- *Inside addresses* — the untranslated addresses of the inside network.
- *Outside addresses* — the addresses that are mapped to the inside addresses.

**When to Use NAT** Use NAT for the following purposes:

- Private address space — You want to connect to the Internet, but your network does not use globally routable IP addresses. If your network uses private addresses, you can use NAT to translate them to access the outside network.
- Load sharing — You want to do load sharing of incoming TCP traffic. For example, traffic destined for a web server identified by one IP address can be redirected to multiple servers with duplicate websites.
- Address migration — You must change your inside addresses. You may change Internet service providers (ISPs), for example, and have to change your numbering scheme. Instead of changing IP addresses on every device in your network, you can use NAT to translate the current addresses into new addresses.
- Address redirection — You want to redirect traffic from one host to another. For example, you want to stop telnet traffic to server A but want to allow traffic to a new server B. Translate the address of server A to that of server B to transparently redirect users to the new server.

**Guidelines** See the following guidelines before configuring NAT:

- Traffic that has embedded IP addresses, such as DNS requests and responses, will not be translated by NAT.
- You should use private IP address space on your inside network recommended by the Internet Assigned Numbers Authority (IANA). The following address ranges are designated as private networks that should not be advertised:  
10.0.0.0 — 10.255.255.255 (10.0.0.0/8)  
172.16.0.0 — 172.31.255.255 (172.16.0.0/12)  
192.168.0.0 — 192.168.255.255 (192.168.0.0/16)  
If you use a different address range that can be validly assigned to someone else's network, you will not be able to communicate with that network. For example, if you FTP from the inside host 1.1.1.1 to a valid outside network host with the IP address 1.1.1.7, the bridge/router will not forward the FTP request outside the network.
- NAT may not be practical if large numbers of hosts in the inside network communicate with the outside network, because NAT is slower than untranslated addresses. Most traffic should originate from or go to hosts within the domain. Because most hosts never communicate with an outside network, only a subset of inside addresses need to be translated into outside addresses.

**Basic NAT Operation** A bridge/router using NAT has at least one port connected to the inside network and one port connected to the outside network. Enable NAT on the port that is directly connected to the outside network. If you have multiple connections to outside networks, enable NAT on each outside port. You must have separate address maps for each NAT port. The inside network is always the network that contains the address you need to map.

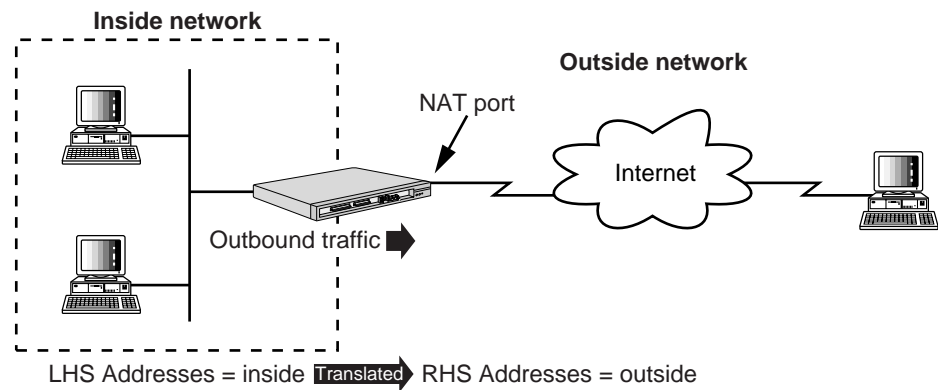
When configuring address maps using the `ADD !<port> -NAT AddressMap` command, you specify the source address (left-hand side (LHS) address) and the translated address (right-hand side (RHS) address). You also specify the direction of the translation: outbound, inbound, bidirectional, or load sharing.

The bridge/router identifies the first packet and the direction of a TCP session and creates a NAT session if a map exists for that direction. However, the bridge/router cannot identify the first packet of a UDP session, so every UDP packet is considered the first packet of a session, which results in a new NAT session for every UDP packet.

**Specifying Direction** Depending on which direction you specify, the LHS addresses can be either the inside or the outside addresses.

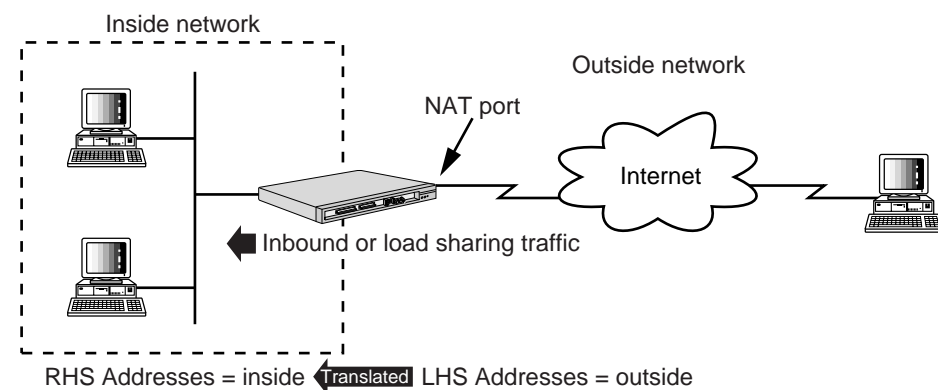
When you specify outbound, the inside addresses are the LHS addresses, which are translated into the outside RHS addresses. Outbound traffic is defined as traffic that leaves the bridge/router through the NAT port.

**Figure 93** Outbound Traffic



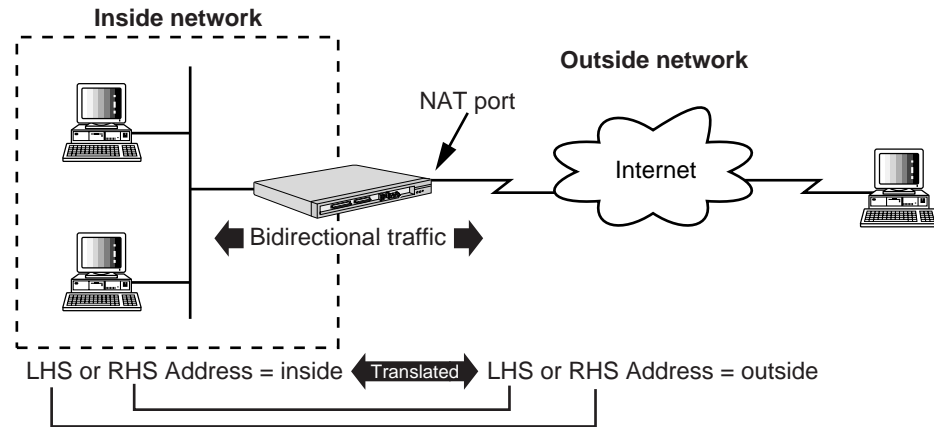
When you specify inbound or load sharing, the LHS addresses are the outside addresses, which are translated into the inside RHS addresses. Inbound or load sharing traffic is defined as traffic that enters the bridge/router through the NAT port.

**Figure 94** Inbound or Load Sharing Traffic



When you specify `bidirectional`, you can use either the LHS or RHS address for the inside address.

**Figure 95** Bidirectional Traffic



## Address Mapping

*Static mapping* establishes a **one-to-one** map between two addresses. Static mapping is useful when a host on the inside network must be reachable by a specific IP address, for example, a web server. Use static mapping with outbound, inbound, or bidirectional connections. Bidirectional translation can use only static mapping.

*Dynamic mapping* establishes a many-to-one, a many-to-many, or a one-to-many map.

A **many-to-one** map, used by outbound or inbound connections, translates multiple addresses into a single address. In an outbound connection, multiple inside hosts can connect using the same outside address because the bridge/router appends a TCP or UDP port number to the outside address for each connection. For example, three hosts are connected at the same time:

| Inside Inside Address | Inside Outside Address:Port |
|-----------------------|-----------------------------|
| 10.0.0.5              | 144.195.23.10:1031          |
| 10.0.0.18             | 144.195.23.10:1032          |
| 10.0.0.25             | 144.195.23.10:1033          |

In an inbound connection, multiple outside addresses are translated into a single inside address.

A **many-to-many** map, used by outbound or load sharing connections, translates multiple addresses by assigning addresses from a pool. Each subsequent connection is assigned the next available address from the pool. If the bridge/router cannot assign an address because it has run out of addresses in the pool, it drops the packet. Make sure you have enough addresses in the address range if you want to use a many-to-many map.

A **one-to-many** map, used by load sharing or outbound connections, translates a single address to one of many addresses. In a load sharing connection, each subsequent connection to the outside address is translated into the next available address from the inside address pool. In an outbound connection, each

subsequent connection from a single inside host is assigned a new IP address, making it appear as though a single user is multiple users.

### NAT Proxy ARP

NAT Proxy ARP is a mechanism that allows you to map addresses for which no actual device exists. NAT address mapping involves mapping addresses in one IP address range to addresses in another IP address range. The translated address could be either one of the addresses assigned to the router interface or it could be a different address. If the translated IP address is different from the directly connected interface address, it is possible that the NAT session setup could fail because no device exists to respond to the initial ARP request.

To allow such configurations, the NAT proxy ARP process checks each NAT mapping entry and adds a static entry into the ARP table. Before the entry is added the following conditions are checked:

- That the added IP address does not belong to the interface's own address.
- That the added IP address is a directly connected address for that port.

The bridge/router then responds to all the ARP requests for which it finds a match in its proxy table with the target IP address in the ARP request. After the ARP request is resolved, the normal NAT session proceeds.

You can view the content of the proxy ARP table by entering:

```
SHow -IP ProxyAddress
```

### Using a Mask

When using the ADD !<port> -NAT AddressMap command, you can specify an address block using a mask: <IPaddr/mask>. <mask> is a number in the range of 0-32, which indicates the number of bits in the IP address that remain unchanged for the IP addresses in that block. The remaining bits in the IP address should be all 0s. The address block includes all addresses except for the first address and the last (x.x.x.255) address.

For example:

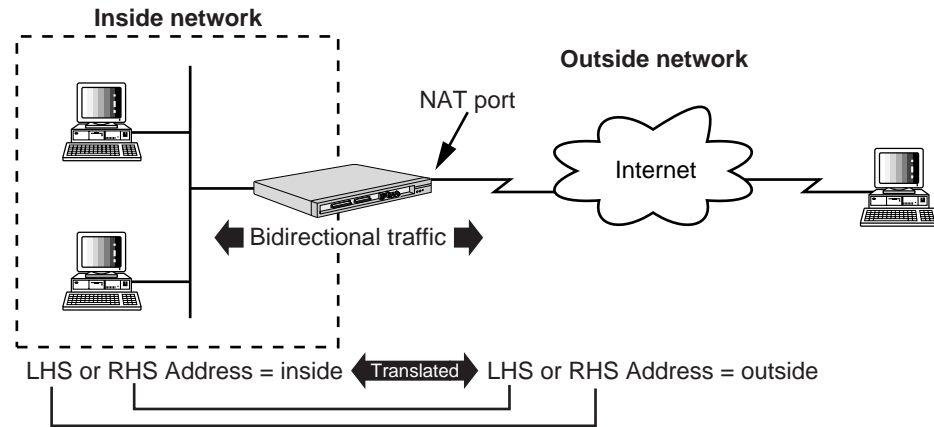
|                |                                                                             |
|----------------|-----------------------------------------------------------------------------|
| 144.195.0.0/16 | All the addresses in the range from 144.195.0.1 to 144.195.255.254          |
| 144.195.1.2/32 | The host itself 144.195.1.2                                                 |
| 0.0.0.0/0      | All the IP addresses in your network                                        |
| 224.0.0.0.1    | All the class D multicast addresses, from 224.0.0.1 through 239.255.255.254 |

### IPCPAddress Mapping

The following example shows how to use IPCPAddress Mapping. If you have Internet connectivity through an Internet Service Provider (ISP) and the ISP provides IP addresses dynamically. This means that the ISP provides different IP addresses every time you dial in.

If you have only one PC then there is no problem. If you have more than one device or a private network as shown in the diagram below, then you must change your NAT mapping every time you dial into the ISP.

**Figure 96** IPCPAddress mapping Network Diagram



You can avoid having to change your NAT mapping every time by using the IPCPAddress Mapping feature.

To configure an OfficeConnect NETBuilder bridge/router to use IPCPAddress mapping, follow these steps:

- 1 Set the IP address to IPCPAddress on port 2. Enter:

```
SETDefault !2 -IP NETAddr = IPCPAddress
```

The router gets the IP address from the ISP through IPCP negotiation and assigns it to port 2.

- 2 Enable NAT on port 2. Enter:

```
SETDefault !2 -NAT CONTROL = Enable
```

- 3 Set port 2 as the default gateway. Enter

```
ADD -IP ROute 0.0.0.0 !2 0
```

The router forwards all IP traffic to the Internet.

- 4 Enable NAT IPCPAddress Mapping on port 2. Enter:

```
SETDefault !2 -NAT IPCPAddressMap = Enable
```

The LHS is set to the default value of 0.0.0.0/0 and the log level is set to Log6. The direction of the mapping will always be set to OutBound. The router creates a NAT Address Mapping on port 2 with the IP address received from the ISP as the RHS address.

For example, after dialing into the ISP you receive the IP address of 123.4.5.6. The IP address for port 2 will be assigned as 123.4.5.6 and a NAT mapping is created on port 2. The NAT mapping is presented as:

```
LHS Address: 0.0.0.0/0
RHS Address: 123.4.5.6
```

Direction : OutBound  
Log : Log6

Now all the devices on the private network are able to access the Internet. After the connection to the ISP is disconnected, the existing NAT sessions remain. These sessions are removed only when the sessions time out or they are deleted. When the connection is restored, the existing NAT sessions are either deleted or not deleted depending on the IP address obtained from the ISP. If you get the same address as before, in this case 123.4.5.6, then the NAT sessions and the NAT Address Mapping are not removed. Otherwise, all the sessions and the mapping are removed and a new NAT Address Mapping is added.

## TCP/UDP Port Mapping

Mapping an address and TCP/UDP port to another address and TCP/UDP port allows more control of the type of traffic NAT translates.

Most TCP and UDP servers use the same port number range, 0-1023, to listen for incoming connections. Most servers use a fixed, well-known port number for listening to a particular service. The major services and their port numbers are listed in Table 24. For a detailed list of reserved services and port numbers, see RFC 1700.

**Table 24** TCP/UDP Port Numbers and Services

| Service | TCP/UDP Port Numbers | Service | TCP/UDP Port Numbers |
|---------|----------------------|---------|----------------------|
| DNS     | 53                   | SMTP    | 25                   |
| finger  | 79                   | SNMP    | 161, 162             |
| FTP     | 20, 21               | syslog  | 514                  |
| Gopher  | 70                   | talk    | 517, 518             |
| HTTP    | 80                   | Telnet  | 23                   |
| NNTP    | 119                  | TFTP    | 69                   |
| NTP     | 123                  | UUCP    | 9540                 |
| POP     | 109, 110             | WAIS    | 210                  |
| RIP     | 520                  | whois   | 43                   |



*TCP/UDP clients use the port number range 1024 - 65535.*

## NAT Scenarios

This section contains the following scenarios:

- Private Address Space
- Load Sharing
- Address Migration
- Address Redirection

### Private Address Space

The following two examples show the same network. Example 1 has the same network defined as the inside network for all steps. Example 2 has opposite networks defined as the inside network for different steps in the same procedure. The inside network is always the network whose addresses need to be mapped.

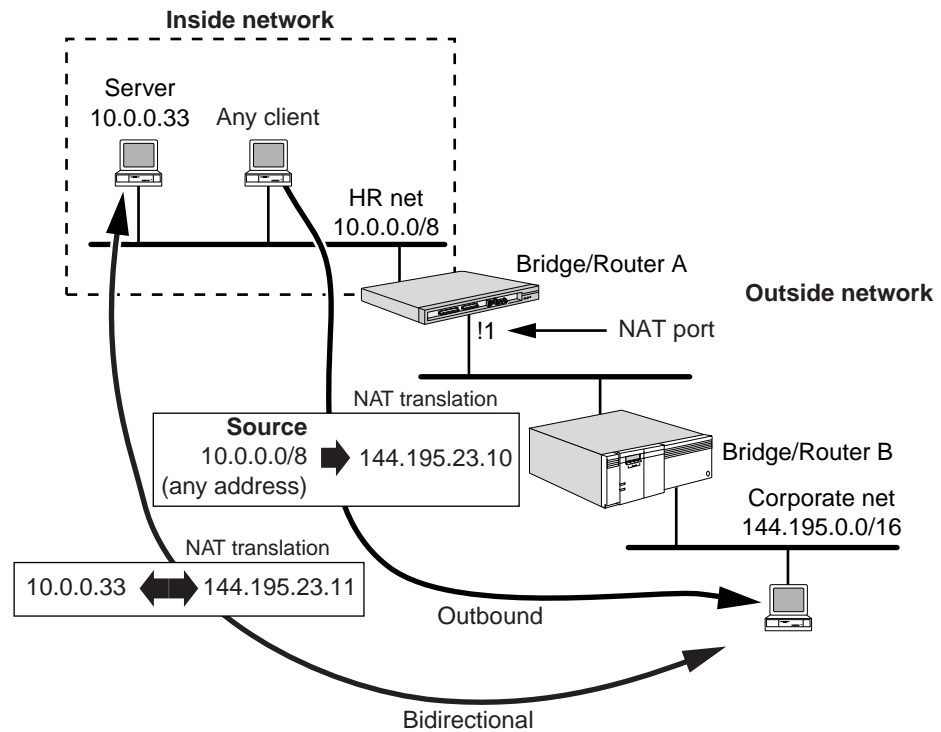
#### Example 1

The Human Resources (HR) network is an isolated network using the address block 10.0.0.0/8, which is a recommended private address block that is not advertised outside the domain.

See Figure 97 and follow these steps to:

- Enable any host in the HR network to connect to the corporate network.
- Allow access to an HR server from the corporate network.

**Figure 97** Private Network Connecting to the Corporate Network



- 1 Enable NAT on port 1 of bridge/router A by entering:

```
SETDefault !1 -NAT CONTROL = Enable
```

The NAT port must be connected to the outside network.

- 2 To access the corporate network from the HR network, map the HR address block 10.0.0.0/8 to a single outside address, 144.195.23.10, and specify outbound by entering:

```
ADD !1 -NAT AddressMap 10.0.0.0/8 144.195.23.10 OutBound
```

Any address from the HR network will be translated into the outside address 144.195.23.10.

- 3 To make the HR server address 10.0.0.33 available to the corporate network, map it to the outside address 144.195.23.11, and specify bidirectional by entering:

```
ADD !1 -NAT 10.0.0.33 144.195.23.11 BiDirectional
```

The bidirectional option allows the server to access the corporate network as well.

*Example 2* The Human Resources (HR) network is an isolated network using the address block 10.0.0.0/8, which is a recommended private address block that is not advertised outside the domain.

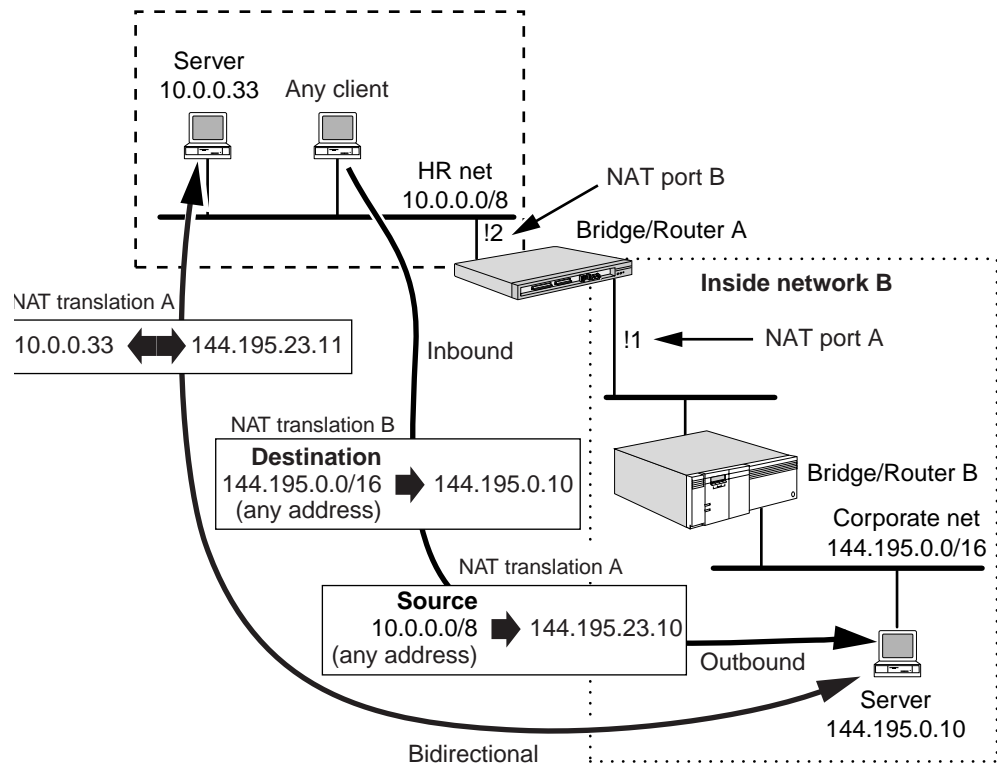
See Figure 98 and follow these steps to:

- Enable any host in the HR network to connect to the corporate network.
- Allow access to an HR server from the corporate network.



- Limit access to the corporate network to only one server.

Figure 98 Relative Inside Networks



- 1 Enable NAT on port 1 of bridge/router A by entering:

```
SETDefault !1 -NAT CONTROL = Enable
```

This NAT port is connected to the outside network with regard to inside network A.

- 2 To access the corporate network from the HR network, map the HR address block 10.0.0.0/8 to a single outside address, 144.195.23.10, and specify outbound by entering:

```
ADD !1 -NAT AddressMap 10.0.0.0/8 144.195.23.10 OutBound
```

Any address from the HR network will be translated into the outside address 144.195.23.10.

- 3 To make the HR server address 10.0.0.33 available to the corporate network, map it to the outside address 144.195.23.11, and specify bidirectional by entering:

```
ADD !1 -NAT 10.0.0.33 144.195.23.11 BiDirectional
```

The bidirectional option allows the server to access the corporate network as well.

- 4 Enable NAT on port 2 of bridge/router A by entering:

```
SETDefault !2 -NAT CONTROL = Enable
```

This NAT port is connected to the outside network with regard to inside network B.

- 5 To limit access from the HR network to only server 144.195.0.10, map all destination addresses in the corporate network to the server address, and specify InBound by entering:

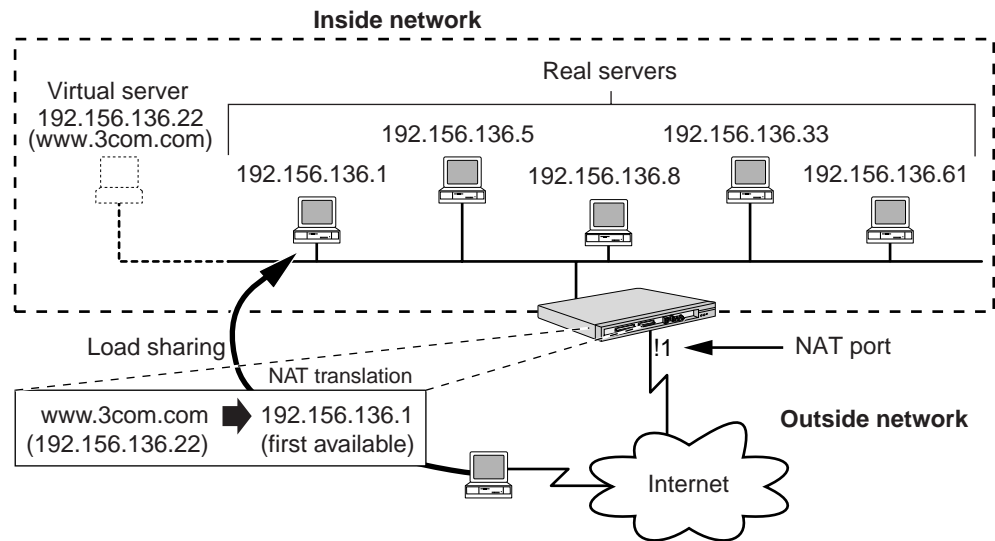
```
ADD !2 -NAT 144.195.0.0/16 144.195.0.10 InBound
```

## Load Sharing

The 3Com web server is replicated on five different servers. The URL `www.3com.com` is accessed thousands of times a day. The domain name server (DNS) advertises the address `192.156.136.22` for `www.3com.com`, even though there is not an actual server at that address.

To forward traffic directed to the virtual server `192.156.136.22` evenly to five web servers, follow these steps (see Figure 99):

**Figure 99** Load Sharing



- 1 Enable NAT on port 1 of the bridge/router by entering:

```
SETDefault !1 -NAT CONTROL = Enable
```

The NAT port must be connected to the outside network.

- 2 To distribute traffic evenly to the five web servers, map the virtual server address `192.156.136.22` to all five server addresses, and specify load sharing by entering:

```
ADD !1 -NAT AddressMap 192.156.136.22 192.156.136.1, 192.156.136.5,
192.156.136.8, 192.156.136.33, 192.156.136.61 LoadShare
```

Load sharing only translates inbound connections. No translation is required for outbound sessions originating from the servers.

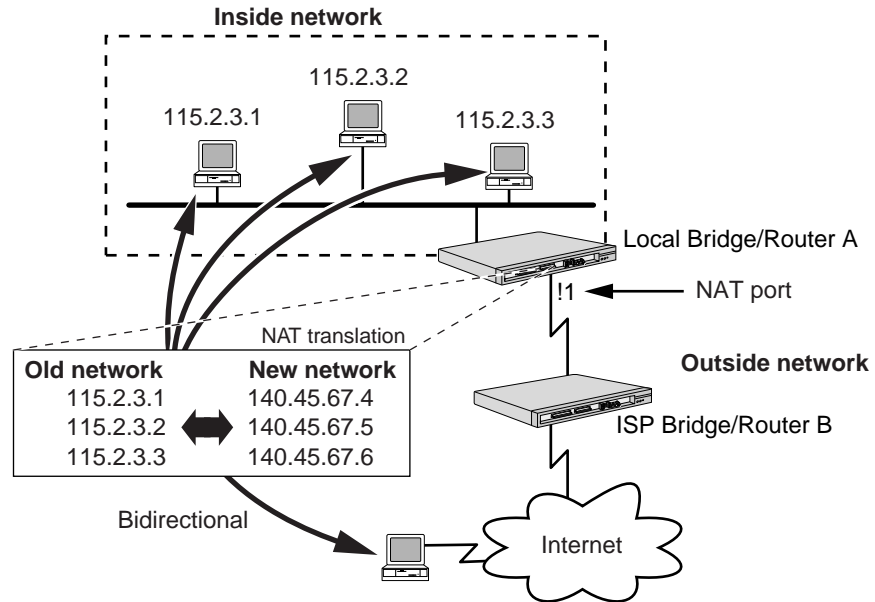
The bridge/router does not detect if a load sharing host is down. However, a log message indicating the possibility that a host may be down is logged. If a host is down, and you want to remove the host from the list, you can delete the original load sharing map (this causes all active sessions to be flushed) and create a new load sharing map without the host.

## Address Migration

Your company has changed ISPs and the new ISP must reassign your IP addresses to work with their network. Reconfiguring every host with a new address requires extensive effort, time, and interruption to users. Moreover, if you change service providers frequently, the process would have to be repeated every time.

To translate each old IP address into a new address, follow these steps (see Figure 100):

**Figure 100** Address Migration



- 1 Enable NAT on port 1 of bridge/router A by entering:

```
SETDefault !1 -NAT CONTROL = Enable
```

The NAT port must be connected to the outside network.

- 2 Map each old address to a new address separately, and specify bidirectional by entering:

```
ADD !1 -NAT AddressMap 115.2.3.1 140.45.67.4 BiDirectional
```

```
ADD !1 -NAT AddressMap 115.2.3.2 140.45.67.5 BiDirectional
```

```
ADD !1 -NAT AddressMap 115.2.3.3 140.45.67.6 BiDirectional
```

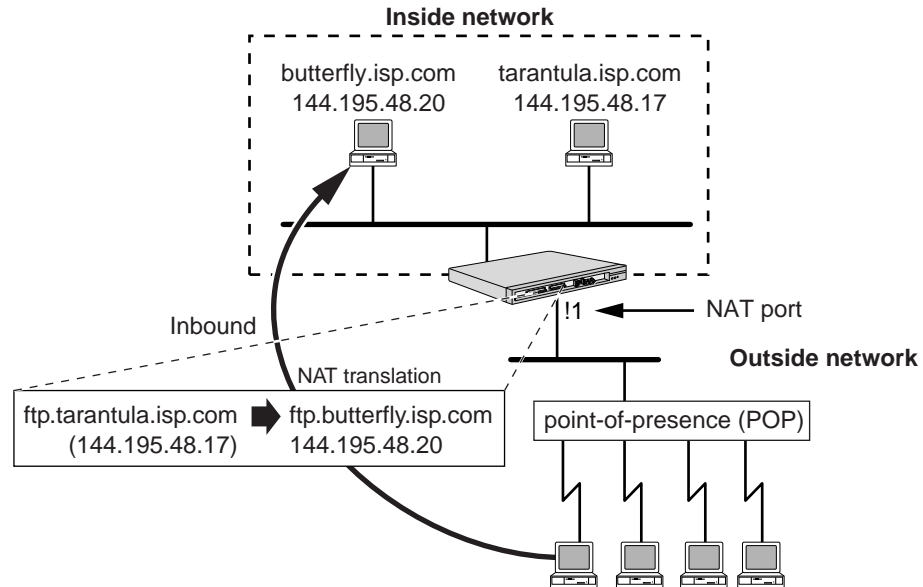
If you do not need bidirectional access, you can map a range of addresses to a new range using the OutBound option.

### Address Redirection

The new, fast FTP server butterfly.isp.com was bought to replace the old FTP server, tarantula.isp.com.

To transparently redirect traffic from tarantula to butterfly, follow these steps (see Figure 101):

**Figure 101** Address Redirection



- 1 Enable NAT on port 1 of bridge/router A by entering:

```
SETDefault !1 -NAT CONTROL = Enable
```

The NAT port must be connected to the outside network.

- 2 Map tarantula (144.195.48.20) to butterfly (144.195.40.17), and specify inbound by entering.

```
ADD !1 -NAT AddressMap 144.195.48.20 144.195.40.17 InBound
```

## IPSEC and NAT

This section describes how NAT will handle an ESP or an AH packet when a NAT router resides in the middle of an IPSEC tunnel.

**Table 25**

| IPSEC Control | Type of Packet | NAT Translation         |
|---------------|----------------|-------------------------|
| Disabled      | ESP            | IP Header, Address Only |
| Disabled      | AH             | IP Header, Address Only |
| Enabled       | ESP            | IP Header, Address Only |
| Enabled       | AH             | No Translation          |

### AH Packet Type

For a given port, NAT does not modify incoming AH packets if IPSEC is also enabled on this port. If IPSEC is disabled, NAT translation is applied. NAT translates AH packets based on the NAT Address Mapping and not on the TcpUdpPort Mapping. None of the data in the AH payload is modified by NAT.

For example, in a TCP packet the TCP port number is not modified by NAT. Similarly, in an ICMP Echo Reply packet NAT does not modify any IP addresses in the payload.

**ESP Packet Type** For a given port, NAT translates all ESP packets whether or not IPSEC is enabled on that port. NAT does not read headers beyond an ESP header because the data beyond the ESP header is encrypted. NAT translates AH packets based on the NAT Address Mapping and not on the TcpUdpPort Mapping. NAT is not able to modify any data beyond the IP header since the data beyond the ESP header is encrypted.

---

### Limitation on One-to-One Mapping

There is a limitation on the one-to-one InBound address mapping. If the LHS address is same as the IP address of the interface, any connection to TCP/UDP services whose port number is less than 1024 will not be translated. NAT does not translate these packets and it will be passed through to the router.

For example:

An InBound NAT Address Mapping for Router port 1:

```
LHS Address: 123.4.5.6
RHS Address: 10.0.0.1
Direction : InBound
IP Address on port 1: 123.4.5.6
```

If the following IP packet is received on port 1 NAT does not translate this packet.

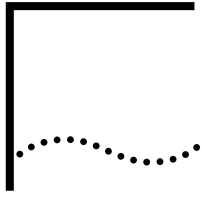
```
IP source address : 123.4.5.7
IP destination address: 123.4.5.6
TCP source port# : 2033
TCP destination port# : 23
```

This packet is received by the router and the client is telnetting into the router and not to 10.0.0.1. To solve this, you must define the following TUP Mapping:

```
LHS Address : 123.4.5.6,23
RHS Address : 10.0.0.1,23
Direction : InBound
```

If you want to map any TCP/UDP services that are less than 1024, you should use the TcpUdpPort Mapping.





# CONFIGURING IP MULTICAST ROUTING

This chapter describes the procedures for configuring your system to perform Internet Protocol (IP) multicast routing. It describes how the multicast router works and gives guidelines for operating, managing, and troubleshooting it.



*For conceptual information, see “Configuring a Multicast Border Router” later in this chapter.*

---

## Configuring a Basic Multicast Router

The procedure in this section describes the minimum number of steps required to configure your system for IP multicasting. Depending on your network requirements, you can use the default values of the parameters in the various services, or you can further configure the router according to later sections in this chapter.

To configure the IP multicast router, you must set parameters in the MIP Service. You must also set parameters in the DVMRP Service if your network uses the Distance Vector Multicast Routing Protocol (DVMRP), or in the MOSPF Service if your network uses the Multicast Open Shortest Path First (MOSPF) routing protocol.

## Configuring for Local Area Networks and Point-to-Point Links

Use this procedure to configure basic IP multicast routing over LAN ports and Point-to-Point Protocol (PPP) links. You can enable both DVMRP and MOSPF if you want to perform route exchanges between the two domains.

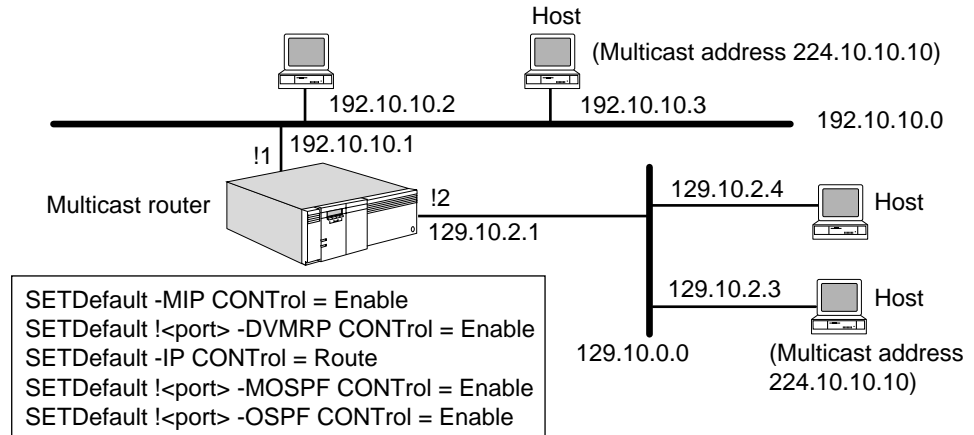
### Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your ports and paths as described in the Configuring Basic Ports and Paths chapter and the Configuring IP Multicast Routing chapter.
- If you are planning to use MOSPF as the multicast routing protocol, you must also set up OSPF for IP unicast routing as described in the Configuring IP Routing chapter.
- Become familiar with the protocols supported by the router. This chapter describes the protocols only when the explanation is necessary for interpreting the parameters and screen displays used in the router software.

### Procedure

To set up a basic configuration, see Figure 102 and follow these steps on the multicast router:

**Figure 102** Configuring Multicast Routing

- 1 Assign an IP address to each router port that will perform IP multicasting using:

```
SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask> [Ones |
Zeros [MTU]]] | UnNumbered
```

Assign the IP addresses to LAN ports and to WAN ports using PPP as the serial line protocol. PPP does not require that you assign an IP address to each wide area port. If you do not want to assign an IP address to a wide area port, you must set the value of the -IP NETaddr parameter to UnNumbered. An advantage of not assigning an IP address to each wide area port is that you conserve valuable network and subnet numbers.

For example, to assign an IP address to port 1 and port 2 of the multicast router in Figure 102, enter:

```
SETDefault !1 -IP NETaddr = 192.10.10.1 255.255.255.0
SETDefault !2 -IP NETaddr = 129.10.2.1 255.255.0.0
```

- 2 Enable the MIP Service by entering:

```
SETDefault -MIP CONTROL = Enable
```

- 3 Determine which multicast routing protocol you want to use.

- Use DVMRP if you want to attach to the Internet Multicast backBONE (MBONE), or if you are not using OSPF as the routing protocol. Complete step a.
- Use MOSPF if you are already running OSPF on your LAN and want multicasting support within an autonomous system. Complete step b.
- If you are connecting to the MBONE, you may also want to use DVMRP. Complete steps a and b.

- a To enable DVMRP, on the each interface using multicast routing, use:

```
SETDefault !<port> -DVMRP CONTROL = Enable
```

- b To enable MOSPF, on the each interface participating in multicast routing, use:

```
SETDefault !<port> -MOSPF CONTROL = Enable
```

The MOSPF Protocol depends on the OSPF Protocol for proper operation. In order to use MOSPF, you must first ensure OSPF is operating correctly. For more information how to enable OSPF, see the Configuring IP Multicast Routing chapter.



You may need to set additional parameters to complete the configuration for PPP. For more information, see the *Configuring Wide Area Networking Using PPP* chapter.

### Configuring for Wide Area Networks

To configure multicast routing using DVMRP over Frame Relay or X.25, see “Configuring DVMRP Multicasting over Frame Relay” or “Configuring DVMRP Multicasting over X.25” later in this chapter.

To configure multicast routing using DVMRP or MOSPF over SMDS, see “Configuring Multicasting over SMDS” later in this chapter.

---

## Verifying the Configuration

This section explains how to verify the status of networks that are reachable from the multicast routers and to get statistics from the router.

### Checking the Overall Status

To check the overall status of your configuration, follow these steps:

- 1 Display the parameter settings in the MIP, DVMRP, and MOSPF Services using:

```
SHow [!<port>] -MIP CONFIguration
SHow [!<port>] -DVMRP CONFIguration
SHow [!<port>] -MOSPF CONFIguration
```

Verify that the MIP, DVMRP, or MOSPF Services are enabled.

- 2 If you are using DVMRP as the multicast routing protocol, verify the entries in the routing table, forwarding table, and the neighboring router table.

- a To display routing and forwarding table entries, use:

```
SHow -DVMRP RouteTable [<subnet>[/<mask>]] [Long]
SHow -DVMRP ForwardTable [<subnet>[/<mask>]] [<group>]
```

In the routing table, check each source subnet and verify that the status is Up.

In the forwarding table, for multicast datagrams that have been sent, make sure there are entries that correspond to the source (source subnet) and destination (multicast group).

For additional information about the displays, see “Controlling the Routing Table” and “Controlling the Forwarding Table”.

- b To display neighboring router information, enter:

```
SHow -DVMRP NeighborRouter
```

Verify that the addresses of neighboring routers appear in the list. For more information about this display, see “NeighborRouter” in the DVMRP Service Parameters chapter in *Reference for Enterprise OS Software*.

- 3 If you are using MOSPF as the multicast routing protocol, verify the entries in the forwarding table by entering:

```
SHow -MOSPF ForwardTable
```

For multicast datagrams that have been sent, make sure there are entries in the forwarding table that correspond to the source and destination.

The MOSPF forwarding table is built only when the router attempts to forward IP multicast packets. The table shows packets the router has recently processed including those successfully forwarded or discarded. The forwarding table varies

from router to router because not all routers have forwarded multicast packets. Routers may periodically flush the forwarding table when topology changes are made.

**4** Examine the local group membership table using:

```
SHow [!<port> | !*] -MIP LocalGroups [<Group addr>]
```

Verify that group memberships of local hosts are displayed in the table.

If you are running MOSPF, only the designated router (DR) and backup designated router (BDR) collect local group information; the table may be empty for non-DR and BDRs.

### Getting Statistics

To view statistics, enter:

```
SHow -SYS STATistics -MIP
SHow -SYS STATistics -DVMRP
SHow -SYS STATistics -MOSPF
```

Statistics for DVMRP and MOSPF reflect data that is forwarded, not control messages. Some control statistics are provided in the MIP Service statistics.

You can collect statistics for a specific period by using the SampleTime and STATistics parameters. For more information on these parameters, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*. For information on interpreting the statistics, see the Statistics Displays appendix.

---

### Troubleshooting the DVMRP Configuration

If you are unable to make connections to other networks after setting up the router, review the following troubleshooting procedure. This procedure can help correct problems in making connections involving more than one multicast router. If the router continues to operate improperly after completing this procedure, contact your network supplier for assistance.

You can find neighboring multicast-capable routers and retrieve status using:

```
MRInfo <target IP> [!<port>] [<timeout (0-120)>]
```

For example, if you are unable to connect to network 128.60.0.0 as shown in Figure 103, on multicast router 4, enter:

```
MRInfo 128.50.0.1
```

The MRInfo command sends an AskNeighbors packet to request neighboring router information. The display provides the addresses of the neighbor, the neighbor's neighbors, the metric, threshold, and status. The status indicates whether the link is down or disabled. It also indicates whether a tunnel link is down, which would prevent multicast router 4 from sending packets to multicast router 1 or to the host on network 128.60.0.0 that listens to multicast address 224.10.10.10.

If no reply packet is received from 128.50.0.1, enter the MRInfo 128.40.0.1 command to see if the tunnel between multicast router 1 and multicast router 2 is up.

You can also discover the multicast tree from a specified receiver to the source using:

```
MTraceRoute <source> <destination> [G <group>] [H <reports>] [!<port>]
[T <timeout>] [W <gateway>] [R <Resp addr>] [L <Resp ttl>]
```

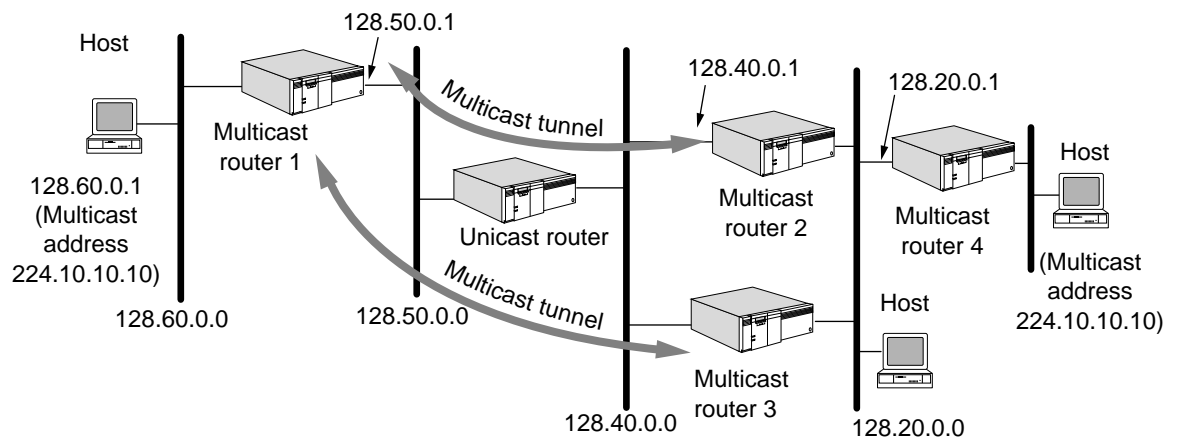
For example in Figure 103, if you want to see the multicast tree from the Host on network 128.60.0.0 to multicast router 4, on multicast router 4, enter:

```
MTraceRoute 128.60.0.1 128.20.0.1
```

Each router in the tree that receives a multicast trace route packet adds its forwarding information associated with the request to the request packet and forwards the packet to the upstream router. When the request packet reaches multicast router 1, multicast router 1 sends a multicast trace route response back to multicast router 4 because the source address is on one of its subnets. The display shows the route from the source to the destination, including hop count, IP subnet, the multicast routing protocol used, threshold, delay time, and error flags.

For more information about the MRInfo and MTraceRoute commands, see the Commands chapter in *Reference for Enterprise OS Software*.

**Figure 103** Troubleshooting Multicast Router Topologies



## Customizing the Multicast Router

After you set up and check the configuration of the basic multicast router, the router begins multicast packet routing among group members. If desired, you can further customize your multicast router as follows:

- Control local group membership queries.
- Adjust the threshold on multicast datagrams.
- Configure multicast routing using DVMRP or MOSPF over SMDS.
- Configure using the DVMRP Protocol.
  - Configure a multicast tunnel for DVMRP routers separated by a nonmulticast router.
  - Configure scoping (filtering) to prevent traffic from being forwarded beyond a boundary router to a set of addresses.
  - Configure multicasting over Frame Relay and X.25.
  - Configure a metric.
  - Control the bandwidth (rate limit) allocated for multicast datagram traffic.

- Configure routing policies.
- Configure forwarding policies.
- Configure route aggregation.
- Control the DVMRP routing and forwarding tables.
- Configure using the MOSPF Protocol.
  - Configure interarea multicast routing.
  - Configure interautonomous (AS) multicast routing.
  - Configure forwarding policies.
  - Display the forwarding table.

### Controlling Local Group Membership Queries

You can control how often Internet Group Management Protocol (IGMP) query messages are sent by the designated router to request local group membership information. For DVMRP routers, the designated router is the router with the lowest IP address. For MOSPF routers, the designated router is the OSPF designated router.

To control local group membership queries, use:

```
SETDefault !<port> -MIP QueryInterval = <seconds>(5-5400)
```

The default setting of this parameter is 120 seconds.

Adjusting the setting of the QueryInterval parameter affects the MembershipExpirationTime, the length of time a local group membership is valid without confirmation. The MembershipExpirationTime is set to two times the value of the QueryInterval parameter plus 20 seconds.

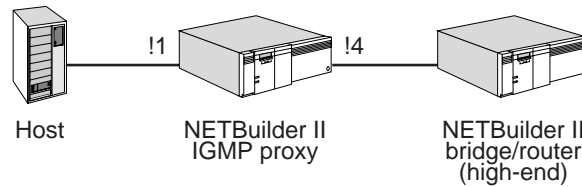
By adjusting the QueryInterval parameter, you control how long entries remain in the local group membership table.

### Configuring an IGMP Proxy Agent and IGMP Version

Enterprise OS devices support IGMP version 2. By default, bridge/routers with Enterprise OS version 11.4 or higher use IGMP version 2 protocol. However, these bridge/routers maintain compatibility with IGMP version 1 to support hosts and local LANs that are running IGMP version 1 protocol.

An IGMP version 2 router may be placed on a subnet where a router on the subnet has not been upgraded to IGMP version 2.

IGMP version 2 also allows configuration of remote/stub routers as IGMP proxy agents. Typically, IGMP proxies are used when a router is not capable of running another multicast routing protocol. Instead of fully participating in multicast routing, these routers will simply forward IGMP message from the host(s) to the upstream multicast router.

**Figure 104** IGMP Proxy Operation

In Figure 104, the bridge/router acts as the proxy for the host. On !1, it becomes an IGMP querier. On !4, it acts as the IGMP Proxy. The IGMP membership queries are sent out on !1 at a regular interval determined by the QueryInterval from the -MIP service parameter. On !4, the proxy sends IGMP membership reports to the high end router (which runs a multicast routing protocol) in response to the IGMP membership query received from the router. The proxy sends the IGMP reports for all the groups learned on its !1 interface. Initially, when the Group membership is learned on !1 for the first time, the proxy sends an unsolicited membership report out on !4. When sending the IGMP reports out on !4, the original source address in the IGMP packet (of the host) is replaced with the !4 address of the proxy.

When the proxy receives the multicast packet on !4, it looks into the list of group addresses learned by it and forwards the packet only on those interfaces where the group membership was reported.

To control IGMP versions and proxy use:

```
SETDefault !<port> -MIP IcmpControl = ([Disable | Enable | Proxy] [V2 | V1])
```

The default settings of this parameter are Disable and V2.

### Adjusting the Multicast Datagram Threshold

You can adjust the threshold on the router to prevent multicast packets whose time-to-live (TTL) value is less than threshold from being forwarded to the given interface. By adjusting the default value, you can provide scope control and prevent certain multicast datagrams from being forwarded out of your network.

To adjust this threshold value, use:

```
SETDefault {!<port> | !<tunnel ID>} -MIP THreshold = <value> (1-255)
```

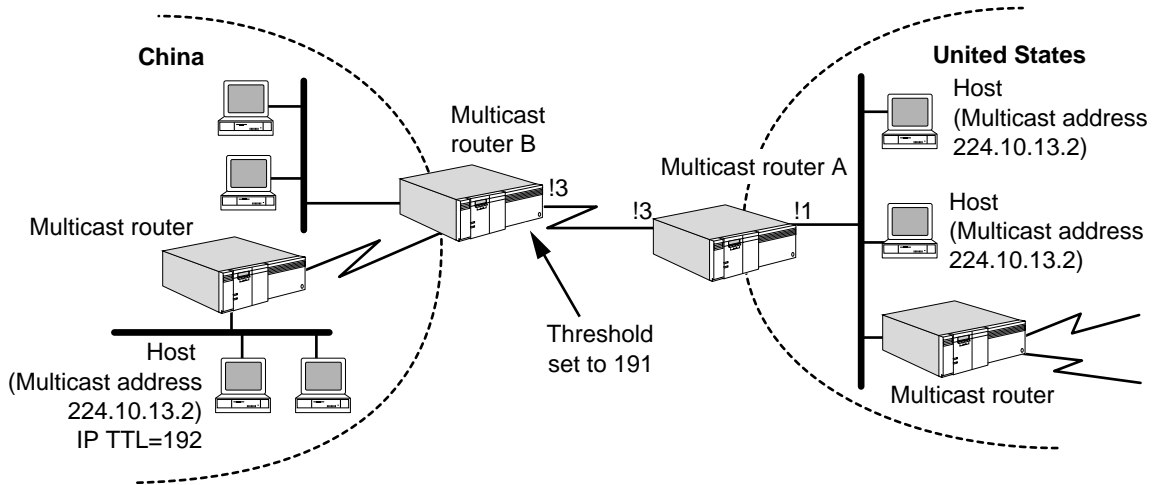
By default, this value is set to 1.

For example, on multicast router B, enter:

```
SETDefault !3 -MIP THreshold = 191
```

Suppose the host in China sends a packet to multicast address 224.10.13.2 with an IP TTL of 192 as shown in Figure 105. As the packet is forwarded by each multicast router in its path, the IP TTL value is decremented. When the packet reaches multicast router B, the IP TTL value is 190. The packet will not be forwarded to multicast router A because packets with a TTL value less than the configured threshold of 191 are prevented from reaching the United States.

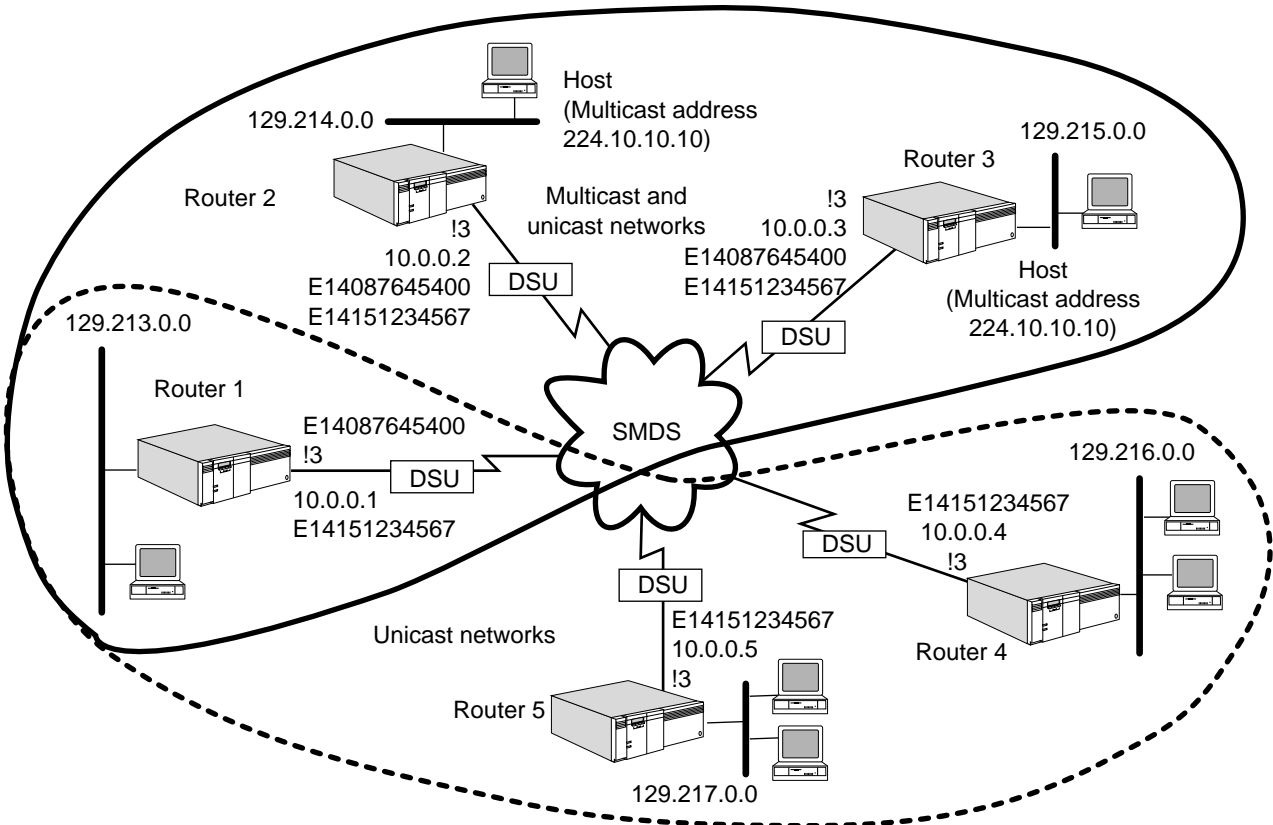
Figure 105 Configuring the Threshold



**Configuring Multicasting over SMDS**

To configure DVMRP or MOSPF multicasting over SMDS, see Figure 106 and follow these steps on both ends of the link:

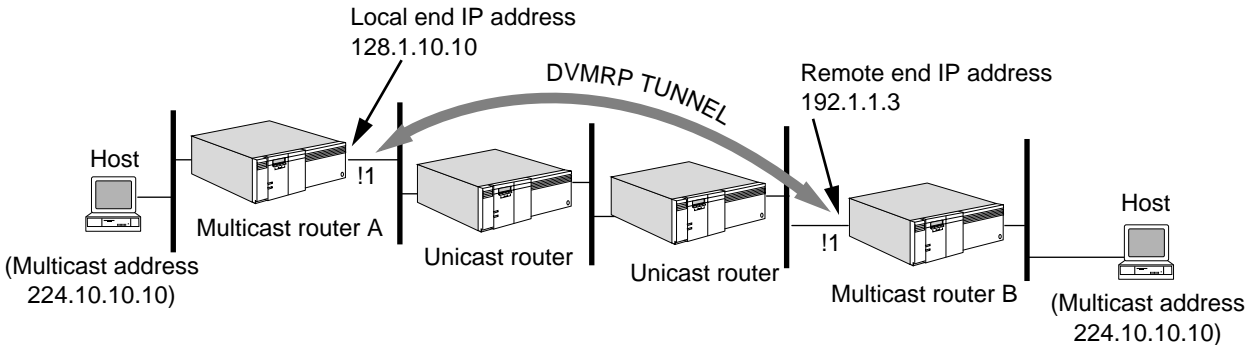
Figure 106 Multicasting over SMDS





To configure a multicast tunnel, see Figure 107 and follow these steps on both ends of the tunnel:

**Figure 107** Configuring a Multicast Tunnel



- 1 Create a virtual point-to-point link between the pair of multicast routers using:

```
SETDefault !<tunnel ID> -DVMRP TUnnel = <local-end IP> <remote-end IP>
 [<t1> (1-255)]
```

For example, on multicast router A, enter:

```
SETDefault !T1 -DVMRP TUnnel = 128.1.10.10 192.1.1.3 3
```

On multicast router B, enter:

```
SETDefault !T1 -DVMRP TUnnel = 192.1.1.3 128.1.10.10 3
```

Up to 32 tunnels can be configured.

The local-end IP address can be any IP address assigned to the system. The remote-end IP address must be unique; you cannot assign tunnels with different local IP addresses and the same remote IP address. The remote-end IP address cannot belong to one of the directly connected subnets if the underlying subnet has broadcast or multicast capability. By default, the TTL is set to 64.

The TTL value should be set to a value greater than or equal to the number of unicast routers in between the multicast routers plus the value of the -MIP Threshold parameter on the remote router interface.



*If IP (unicast) routing is not enabled, you must configure a static route for the remote end of the tunnel.*

- 2 Enable DVMRP routing on the specified tunnel interface using:

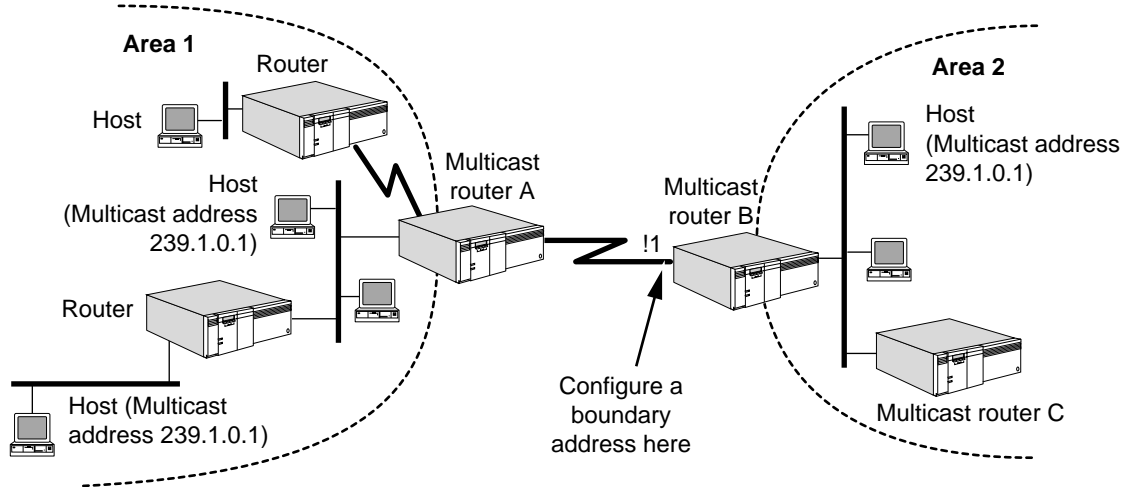
```
SETDefault !<tunnel ID> -DVMRP CONTrol = Enable
```

### Configuring DVMRP Scoping

To configure scoping (filtering), see Figure 108 and configure a set of multicast destinations that are not reachable through the boundary router port or tunnel.



Figure 108 Configuring Scoping



Use:

```
ADD {!<port> | !<tunnel ID>} -DVMRP BoundaryAddr <IP addr> [<subnet mask>]
```

For example, to configure multicast router B with a boundary address so that packets destined to the group of multicast addresses 239.1.0.1 through 239.1.255.1 are dropped, enter:

```
ADD !1 -DVMRP BoundaryAddr 239.1.1.1 255.255.0.255
```

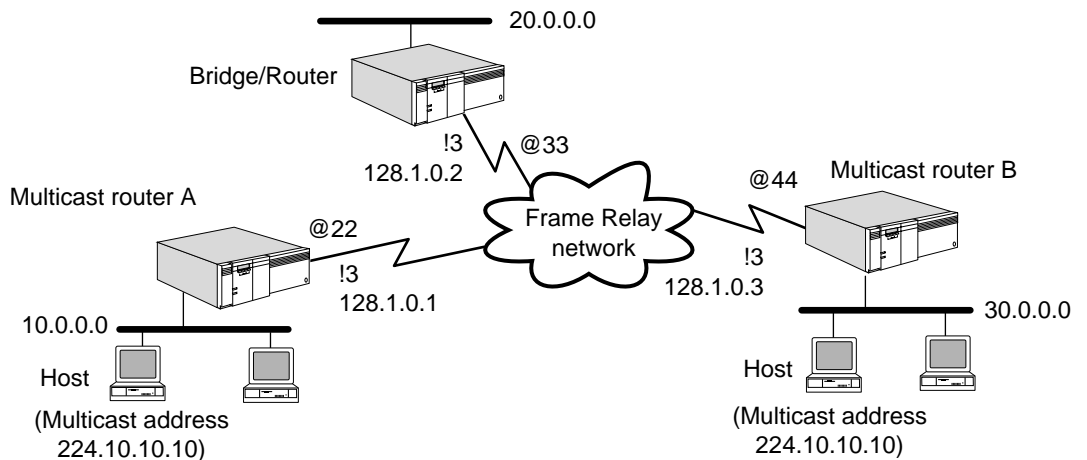
Packets from area 1 destined to the above address ranges do not reach area 2; packets from area 2 destined to one of the blocked address also do not reach area 1.

You can block a single address by not specifying the subnet mask (the default subnet mask is 255.255.255.255).

### Configuring DVMRP Multicasting over Frame Relay

To configure multicasting over Frame Relay, see Figure 109 and follow these steps on both ends of the link:

Figure 109 Multicasting over Frame Relay



- 1 Set up the Frame Relay Service as described in "Setting Up the Frame Relay Service" in the Configuring Wide Area Networking Using Frame Relay chapter.

- Assign an IP address to each router wide area port that will perform IP multicasting using:

```
SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask> [Ones | Zeros [MTU]]]
```

- Add a neighbor address over the Frame Relay network using:

```
ADD !<port> -DVMRP NEighbor <FR_DLCI>
```

Specify the Frame Relay data link connection identifier (DLCI) address associated with the permanent virtual circuit.

For example, only multicast routers A and B are participating in multicast routing. To configure the neighbor address, on multicast router A, specifying the DLCI address of multicast router B, enter:

```
ADD !3 -DVMRP NEighbor @44
```

On multicast router B, enter the same command and specify the DLCI address of multicast router A.

- Enable the DVMRP routing protocol on each wide area port using:

```
SETDefault !<port> -DVMRP CONTrol = Enable
```

- Display neighboring router information using:

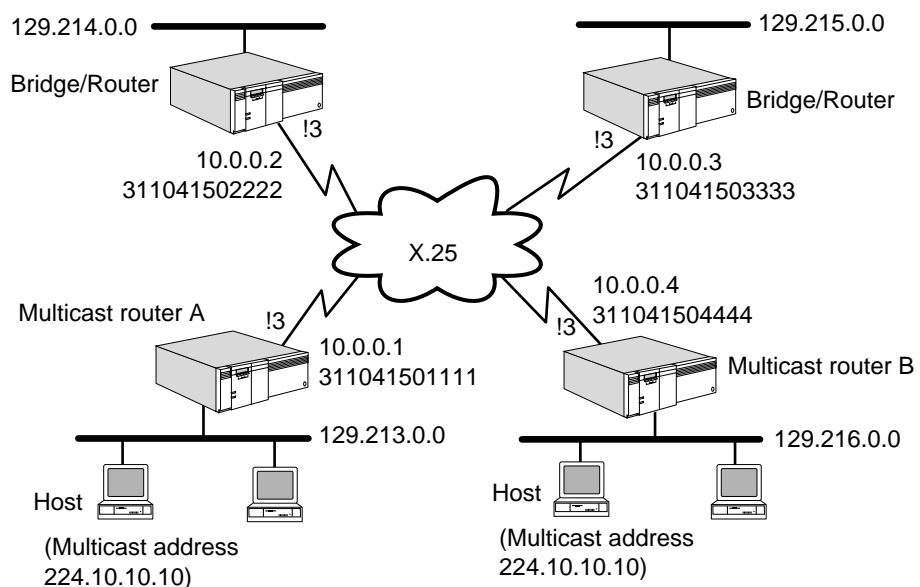
```
SHow !<port> -DVMRP NeighborRouter [<IP addr>]
```

If <IP addr> is specified, only neighboring router information for this IP address is displayed. For more information about elements in the display, see "NeighborRouter" in the DVMRP Service Parameters chapter in *Reference for Enterprise OS Software*.

### Configuring DVMRP Multicasting over X.25

To configure multicasting over X.25, see Figure 110 and follow these steps on both ends of the link:

**Figure 110** Multicasting over X.25



- 1 Set up the X25 Service as described in "Setting Up the X25 Service" in the Configuring Wide Area Networking Using X.25 chapter.
- 2 Assign an IP address to each router wide area port that will perform IP multicasting using:

```
SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask> [Ones | Zeros
[MTU]]]
```

- 3 Add neighbor address over the X.25 network using:

```
ADD !<port> -DVMRP NEighbor <X.25 DTE>
```

Specify the X.25 address associated with the remote router.

For example, only multicast routers A and B are participating in multicast routing. To configure the neighbor address, on multicast router A, specifying the DTE address of multicast router B, enter:

```
ADD !3 -DVMRP NEighbor #311041504444
```

On multicast router B, enter the same command and specify the DTE address of multicast router A.

- 4 Enable the DVMRP routing protocol on each wide area port using:

```
SETDefault !<port> -DVMRP CONTrol = Enable
```

- 5 Display neighboring router information using:

```
SHow !<port> -DVMRP NeighborRouter [<IP addr>]
```

If <IP addr> is specified, only neighboring router information for this IP address is displayed. For more information about elements in the display, see "NeighborRouter" in *Reference for Enterprise OS Software*.

### Configuring a DVMRP Metric

You can configure a metric, or administrative cost, on an interface using:

```
SETDefault {!<port> | !<tunnel ID>} -DVMRP METric = <value> (1-31)
```

The default metric is 1.

You may want to adjust the metric if you have multiple routes to the same source and want one route selected over the other. For example, suppose that DVMRP learns about two routes to the same source. Route 1 has an administrative cost of 25; Route 2 has an administrative cost of 3. The DVMRP Protocol selects Route 2 because it is the route with the lowest metric.

### Controlling the DVMRP Rate Limit for Multicast Traffic

The DVMRP rate limit is the bandwidth measured in kilobits per second. You can control the rate limit that is allocated for multicast datagram traffic using:

```
SETDefault {!<port> | !<tunnel ID>} -DVMRP RateLimit = <Kbits/second>
(0-100000)
```

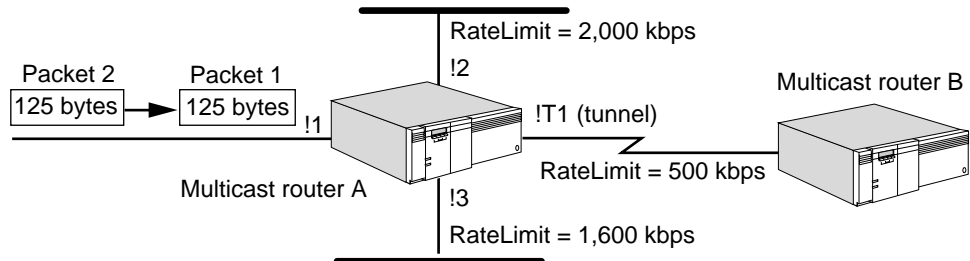
The default is 0, which means that no limit is applied to the given interface, and the interface uses its full bandwidth.

To set the rate limit on multicast router A in Figure 111, enter:

```
SETDefault !2 -DVMRP RateLimit = 2000
SETDefault !3 -DVMRP RateLimit = 1600
SETDefault !T1 -DVMRP RateLimit = 500
```

To control your multicast traffic, you need to configure the rate limit if you are connected to the MBONE, which anticipates traffic at a rate of 500 kbps. See Figure 111 and the explanation that follows.

**Figure 111** Controlling the Rate Limit



When multicast router A in Figure 111 receives two 125-byte packets, it queues the packets into the ports' transmit queues (because of the rate limit settings) instead of immediately forwarding them. Multicast router A controls packet forwarding as follows:

After 1 millisecond, port 2 assigns tokens at a rate limit of 2,000 kilobits per second (kbps) (2,000 bits per millisecond or 250 bytes per millisecond). The router extracts both packets from port 2's transmit queue and forwards them to port 2's attached LAN.

Port 3 assigns tokens at a rate limit of 1,600 kbps (1,600 bits per millisecond or 200 bytes per millisecond). The router can forward only the first 125-byte packet on the attached LAN after 1 millisecond. After the next millisecond, port 3 receives another 200 byte token and can transmit up to 275 bytes (200 - 125 + 200); therefore, the router forwards packet 2 (175 bytes) on port 3's attached LAN.

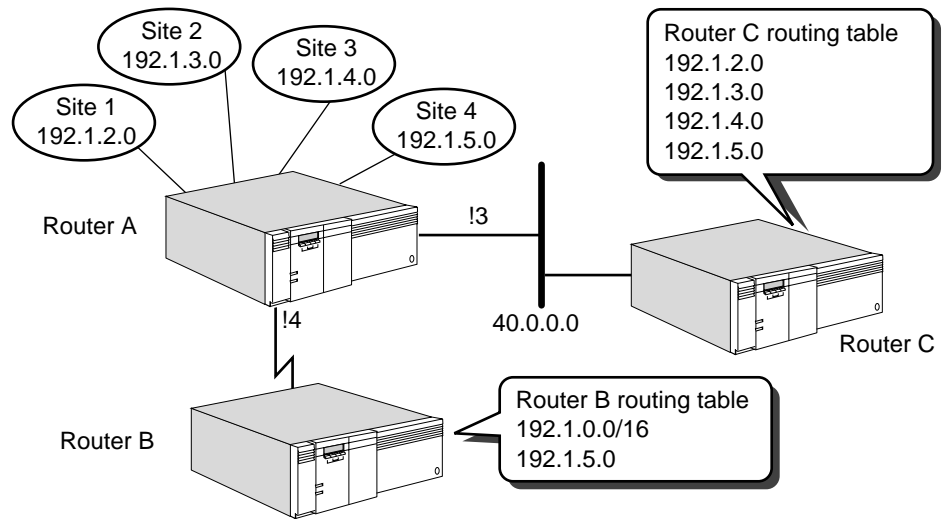
The tunnel interface (!T1) assigns tokens at a rate limit of 500 kbps (500 bits per millisecond or 62.5 bytes per millisecond). After 2 milliseconds, the router forwards packet 1 (125 bytes) on the tunnel interface. After 4 milliseconds, the router forwards packet 2 (175 bytes) on the tunnel interface.

### Configuring DVMRP Route Aggregation

With DVMRP route aggregation, you can combine the characteristics of several different routes so that a single route can be advertised. By combining several networks into one supernet, the number of route report messages and the size of the routing table are reduced.

To configure route aggregation, see Figure 112 and follow these steps:

**Figure 112** DVMRP Route Aggregation



- 1 Specify a list of networks that DVMRP advertises as a single supernet route using:

```
ADD -DVMRP AggregateRange <subnet>/<mask> [<metric>]
```

For example, to combine the routes to sites 1, 2, 3, and 4 into a range so that only a single route is advertised on router A, enter:

```
ADD -DVMRP AggregateRange 192.1.0.0/16
```

- 2 Specify a list of routes that DVMRP explicitly advertises using:

```
ADD -DVMRP AggregateExcept <subnet>/<mask>
```

For example, if you do not want site 4 included in the aggregation range, enter:

```
ADD -DVMRP AggregateExcept 192.1.5.0/24
```

- 3 Enable route aggregation and the DVMRP routing protocol by entering:

```
SETDefault !4 -DVMRP CONTROL = (Enable, Aggregate)
```

As shown in Figure 112, router A advertises a single network (192.1.0.0/16) that summarizes each of the three connected sites and also explicitly advertises the exception route (192.1.5.0) to router B. Without the use of aggregation, router A advertises each route with a separate entry as shown in the router C routing table, which grows in size. With route aggregation, the router B routing table has an entry for 192.1.0.0 and 192.1.5.0.

### Controlling the Routing Table

You can control how often the router sends route report messages, delete entries in the routing table, and display the routing table.

To control how often the router sends route report messages containing the complete routing table, use:

```
SETDefault -DVMRP UpdateTime = <seconds>(5-5400)
```

By default, DVMRP updates the routing table every 60 seconds. By changing this setting, you affect how long a route is considered valid (*RouteExpirationTime*) and how long a route exists without confirmation (*GarbageCollectionTime*). The

RouteExpirationTime is equal to three times the value of this parameter, and the GarbageCollectionTime is equal to five times the value of this parameter. By increasing the value of the UpdateTime parameter, you can reduce the amount of route report traffic but you may also increase the size of the routing table.

This parameter can determine how long a neighbor is considered “up” without confirmation (NeighborExpireTime) and when to consider the associated virtual interface as a leaf link (LeafConfirmationTime). The NeighborExpireTime is set to two times the value of this parameter plus 20 seconds, and the LeafConfirmationTime is set to three times the value of this parameter plus 20 seconds.

To flush entries in the routing table learned from DVMRP, use:

```
FLush -DVMRP RouteTable
```

To display the routing table, use:

```
SHow -DVMRP RouteTable [<subnet>[/<mask>]] [Long]
```

If the <subnet> and/or <mask> syntax is specified, the routing table for the range of specified subnets is displayed. If Long is specified, the display shows a lists of ports that connect to child subtrees and leaf subnets.

For example, to display the following table, enter:

```
SHow -DVMRP RouteTable Long
```

| SourceSubnet | SubnetMask | FromGateway | Metric | Status | TTL | InPort | OutPorts    |
|--------------|------------|-------------|--------|--------|-----|--------|-------------|
| 20.0.0.0     | 255.0.0.0  | 11.11.11.11 | 3      | Up     | 200 | 1      | 2, 3*, 4, 5 |
| 30.0.0.0     | 255.0.0.0  | ----        | 0      | Up     | --  | 2      | 1, 3*, 4, 5 |
| 40.0.0.0     | 255.0.0.0  | 11.11.11.11 | 6      | GC     | 100 | 1      | 2, 3*, 4    |

The display consists of the following items:

|              |                                                                                                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SourceSubnet | The original subnet from which the multicast datagram originated.                                                                                                                                               |
| SubnetMask   | The subnet mask of the source subnet.                                                                                                                                                                           |
| FromGateway  | The previous hop router that leads back to the source. If no gateway is specified, the subnet is directly connected.                                                                                            |
| Metric       | The routing metric of the path back to the source subnet.                                                                                                                                                       |
| Status       | The status of this route entry: Up, GC (Garbage-Collect), HD (Hold-Down), and Down.                                                                                                                             |
| TTL          | Time-to-live indicates how much time (in seconds) is left before removing an entry from routing table.                                                                                                          |
| InPort       | Incoming port for the multicast datagrams from that source.                                                                                                                                                     |
| OutPorts     | List of ports on which multicast datagrams originated from this source are forwarded. An asterisk (*) indicates that the outgoing port connects to a leaf of the multicast delivery tree rooted at this source. |

### Controlling the Forwarding Table

You can specify how long you want to keep a (source, group) pair in the forwarding table and display the contents of this table.

To control how long entries remain in the forwarding table, use:

```
SETDefault -DVMRP CacheTime = <seconds> (300-86400)
```

The default value of this parameter is 300 seconds. You can adjust the setting up to 1 day (86,400 seconds). By adjusting the CacheTime parameter, you can control the size of the forwarding table.

To display entries in the forwarding table, use:

```
SHoW -DVMRP ForwardTable [<subnet>[/<mask>]] [<group>]
```

You can display the current table for each (source, group) pair. If you specify only the subnet, all group entries associated with this subnet are displayed. If you specify only the group, all source subnets associated with this group are displayed. If you specify both the subnet and group, only this particular entry is displayed.

For example, to display the following table, enter:

```
SHoW -DVMRP ForwardTable
```

| SourceSubnet | MulticastGroup | TTL | InPort | OutPorts |
|--------------|----------------|-----|--------|----------|
| 20.0.0.0     | 224.1.1.1      | 200 | 1 Pr   | 2p 3p 4p |
|              | 224.2.2.2      | 100 | 1      | 2p 3 4   |
|              | 224.3.3.3      | 250 | 1      | 2 4b     |
| 30.0.0.0     | 224.1.1.1      | 300 | 1      | 2 3 4    |
|              | 239.4.4.4      | 100 | 1 Sc   |          |

The display consists of the following items:

|                |                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| SourceSubnet   | The original subnet of multicast datagrams.                                                                                              |
| MulticastGroup | The group address to which multicast datagrams are destined from the origin.                                                             |
| TTL            | Time-to-live indicates how much time (in seconds) are left before removing a source and group entry from the table.                      |
| InPort         | Indicates the incoming port for the multicast datagrams from that source.                                                                |
|                | Pr A Prune message is sent to the upstream router.                                                                                       |
|                | Sc The multicast group address is configured as a boundary address, and no traffic for this group address is forwarded from that port.   |
| OutPorts       | Indicates the ports that multicast datagrams belonging to this group are forwarded.                                                      |
|                | p The port receives all the Prune messages of the downstream neighboring routers, and no multicast datagrams are forwarded to this port. |
|                | b The multicast group address is configured as a boundary address, and no traffic for this group address is forwarded to this port.      |

### Using the MOSPF Protocol

The following sections describe how to further customize your multicast router if you are using the MOSPF Protocol as the multicast routing protocol.

## Configuring Interarea Multicasting

To perform interarea multicasting when running the MOSPF Protocol, the Area Border Router (ABR) must be configured as an interarea multicast forwarder, which is an ABR with multicast extensions enabled.

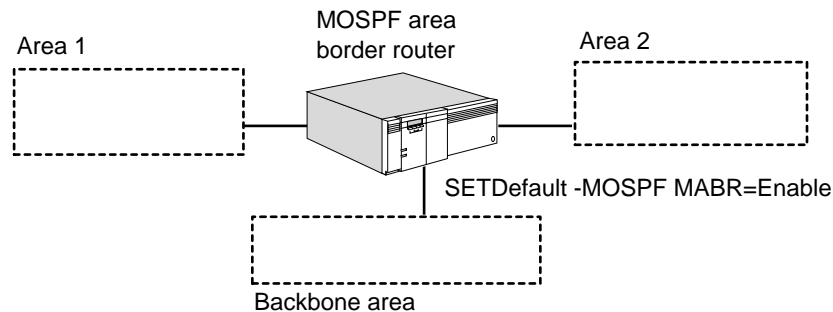
As shown in Figure 113, the ABR connects two areas to the backbone. The ABR must be configured as an interarea multicast forwarder so that it can summarize group membership information from attached nonbackbone areas into the backbone and to forward multicast packets between areas.

To allow multicasting between areas, enter:

```
SETDefault -MOSPF MABR = Enable
```

The router must be an OSPF ABR for the MABR parameter to take effect. By default, this parameter is enabled.

**Figure 113** Interarea Multicasting



For more information, see “Interarea Multicasting” later in this chapter.

## Configuring MOSPF Routing Policies

Using the routing policies supported by MOSPF, you can control the reporting of routes learned from other sources for interautonomous system multicasting. The current implementation of MOSPF routing policies only supports DVMRP as the source of multicast traffic.

- 1 Enable DVMRP routing information to be advertised into the MOSPF domain using:

```
ADD -MIP RoutePolicy from <DVMRP | MOSPF | PIM> to <DVMRP | MOSPF | PIM>
 <Subnet>/<mask> [Aggregate | Individual | Reject] [<metric>] [Type1 |
 Type2]
```

Supply the subnet and mask of the address range of the DVMRP route to be advertised. The mask value is the number of leading 1s in the mask and ranges from 0 to 32.

The <subnet>/<mask> describes a range of addresses. For example:

- 10.0.0.0/8 describes all the subnets within network 10.
- 10.1.0.0/16 describes all the subnets within 10.1.0.0.
- 0.0.0.0/0 describes all subnets.



An address can fall into multiple subnet/mask ranges. In this situation, the range with the highest mask bits is chosen. The range 0.0.0.0/0 is always the lowest priority.

The keyword `Aggregate` means that MOSPF advertises a single subnet/mask route, which can summarize multiple networks into a single network. The keyword `Individual` means that all individual source subnets are accepted and advertised as learned into the MOSPF domain. The keyword `Reject` means the specified source network is rejected (not advertised).

You can optionally supply a metric value from 0 to 65,535.

You can select either `Type1` or `Type2`. `Type1` advertises the routes as a type 1 external LSA, which is always preferred over a type 2 external LSA for the same destination.

For example, to accept and aggregate routes from 192.10.10.0 advertised as a single route into the MOSPF domain, enter:

```
ADD -MIP RoutePolicy from DVMRP 192.10.10.0/24 Aggregate
```

To accept and advertise all routes learned from 129.213.0.0 sourced from the DVMRP domain, enter:

```
ADD -MIP RoutePolicy from DVMRP 129.213.0.0/16 Individual
```

To reject all other routes, including transmissions from the MBONE, enter:

```
ADD -MIP RoutePolicy from DVMRP 0.0.0.0/0 Reject
```

- 2 Enable the MOSPF router to perform interautonomous system multicast forwarding by entering:

```
SETDefault -MIP ForwardControl from MOSPF to DVMRP Enable
```

When this command is executed, the MOSPF router declares itself as a wild-card multicast receiver to all its attached areas to attract multicast packets to all destinations. It imports specified routes sourced from DVMRP into the MOSPF routing domain as external LSAs.

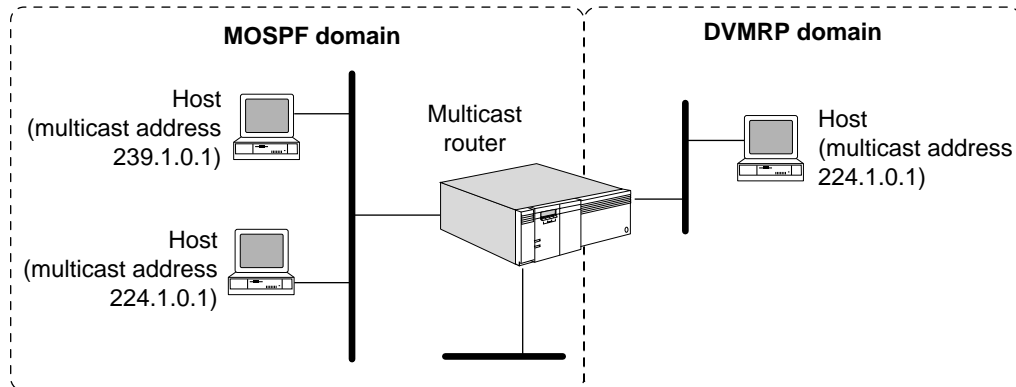
This parameter only enables the MOSPF domain to accept DVMRP-sourced multicast packets. For the DVMRP domain to accept MOSPF-sourced multicast packets, see "Using the DVMRP Protocol" earlier in this chapter. Failure to configure the DVMRP routing policies results in half-duplex communication.

### Configuring MOSPF Forwarding Policies

Using MOSPF forwarding policies, you can filter destination groups and control data packet forwarding between MOSPF and DVMRP domains.

To configure your MOSPF router for destination group filtering, see Figure 114 and follow these steps on the multicast router:

**Figure 114** MOSPF Destination Group Filtering



- 1 Configure a list of destination group addresses whose data packets are accepted and forwarded, or rejected and dropped, using:

```
ADD -MIP ForwardPolicy from <DVMRP | MOSPF | PIM> to <DVMRP | MOSPF | PIM>
<SubnetPrefix>/<mask> [Accept | Reject]
```

The <subnet>/<mask> syntax describes a range of addresses to either be accepted or rejected by MOSPF. For example:

- 239.0.0.0/8 describes all the addresses within network 239.
- 239.1.0.0/16 describes all the addresses within network 239.1.
- 239.1.10.0/24 describes all the addresses within network 239.1.10.

The Accept option causes the following actions by the multicast router:

- If the multicast router receives a packet from the DVMRP domain with a destination address that matches this destination group filter, then the multicast router accepts it and forwards it into the MOSPF domain;
- If the multicast router receives a packet from the MOSPF domain with a destination address that matches this destination group filter, then the multicast router accepts it and forwards it into the DVMRP domain.

The Reject option causes the following actions by the multicast router:

- If the multicast router receives a packet from the DVMRP domain with a destination address that matches this destination group filter, then the multicast router rejects it and drops the packet and never forwards it into the MOSPF domain.
- If the multicast router receives a packet from the MOSPF domain with a destination address that matches this destination group filter, then the multicast router rejects it and drops the packet and never forwards it into the DVMRP domain.

For example, to configure data packets to the destination group 224.1.0.1 to be rejected on the multicast router, enter:

```
ADD -MIP ForwardPolicy from DVMRP to MOSPF 224.1.0.0/16 Reject
```

- 2 Enable the policy using:

```
SETDefault -MIP ForwardControl from <DVMRP | MOSPF | PIM> to <DVMRP |
MOSPF | PIM> <DestGroup | NoDestGroup>
```

Data packets to destination group addresses 224.1.0.0 to 224.1.255.255 between DVMRP and MOSPF domains are rejected and dropped by the multicast router.

## Configuring a Multicast Border Router

The Multicast Border Router (MBR) functionality provides for the efficient interoperation among independent multicast routing protocols, such as DVMRP, MOSPF, and PIM-SIM. The MBR allows sources and receivers inside multiple autonomous multicast routing domains (each running a different multicast routing protocol) to communicate. These domains must be connected by MBRs where the primary role of the MBR is to forward the traffic from one domain to another.

An MBR consists of two or more active multicast routing interfaces, each running an instance of some multicast routing protocol, such as DVMRP or PIM. The bridge/router is configured to forward packets between two or more independent multicast domains. Only one multicast protocol can be active on each interface.

The ForwardControl parameter in the Enterprise OS -MIP Service controls the forwarding of packets between the multicast routing domains on the MBR.

To control the forwarding between multicast routing domains, use:

```
SETDefault -MIP ForwardControl from <DVMRP | MOSPF | PIM> to <DVMRP | MOSPF | PIM> Enable | Disable
```

When this parameter is set to DestGroup, data packets are forwarded between the "from" and "to" domains according to the lists established by the -MIP Service ForwardFilter parameter.

When a bridge/router has DVMRP, MOSPF, and PIM interfaces to forward data packets between the DVMRP and PIM domains, enter:

```
SETDefault -MIP ForwardControl from DVMRP to PIM Enable
```

To control the routing information exchange between the multiple protocols, use:

```
Add -MIP RouteFilter <DVMRP | MOSPF | PIM> to <DVMRP | MOSPF | PIM> <GroupPrefix>/<mask> [Aggregate | Individual | Reject] [<metric>] [Type 1 | Type 2]
```

For example, to accept routes from 192.10.10.0 advertised as a single route into the MOSPF domain from the DVMRP domain, enter:

```
ADD -MIP RouteFilter from DVMRP to MOSPF 192.10.10.0/24 aggregate
```

## How the IP Multicast Router Works

IP Multicast is a network service in which a single IP packet can be received by multiple hosts. Using this service, an application can send one copy of each packet and address it to a group of computers that want to receive it.

IP Multicast is a more efficient way to use network resources, especially for bandwidth-intensive services like audio and video applications.

For example, using regular IP (unicast) to send an audio and video conference lecture from one computer to multiple hosts would result in a set of individual packet streams, one each from the originating computer to the receiving

computers. This large amount of network traffic could overload network segments and packet switches.

If IP multicast packets are used instead, then the transmitting computer generates only a single stream of multicast packets that is sent to a multicast session address. The multicast routing on campus will direct this packet stream to only those subnets on campus where some computer(s) have indicated an interest in receiving the multicast session.

All receiving computers program their network interface to listen for packets sent to the specific multicast session address that is associated with the conference lecture. Even if there are many listeners on a single Ethernet LAN, only one multicast packet stream is transmitted on the LAN for that session address. All receiving computers listening to that multicast session receive the same packet stream, which can result in a large reduction in the amount of network traffic that must be sent to the receiving LAN.

Membership in a multicast group is dynamic. A host may join or leave a group at any time. A host may be a member of an arbitrary number of multicast groups; group members can span multiple subnets. A host may send datagrams to a multicast group without being a member.

Each multicast group has a unique multicast (Class D) address. Some multicast addresses are assigned by the Internet Addressing and Naming Authority (IANA) and correspond to groups that always exist even if they have no current members. Such addresses are said to be well-known. Typically, packets transmitted to these addresses use a TTL of 1. Other multicast addresses are available for temporary use. They correspond to transient multicast groups that are created when needed and discarded when the membership reaches zero. For more information, see "Multicast Addresses" later in this chapter.

Special gateways, or routers, forward multicast packets, but hosts do not need to explicitly know about these routers. It is the responsibility of the multicast router to receive the multicast packet from the host and correctly forward it to those members of the group.

Hosts and routers must run the IGMP Protocol for multicast connectivity. In addition, the router must run one or more of the following multicasting routing protocols:

- DVMRP
- MOSPF Version 2
- Core-Based Trees (CBT)
- Protocol Independent Multicast (PIM) (Sparse and/or Dense Mode)

The Enterprise OS software includes the DVMRP and MOSPF routing protocols, which are user configurable, and the IGMP Protocol, which requires no configuration. For more information, see "Distance Vector Multicast Routing Protocol," "Multicast Open Shortest Path First Protocol," and "Internet Group Management Protocol" later in this chapter.

### **MBONE Connectivity with Multicasting**

The MBONE is a virtual network running on top of the Internet that is composed of a cooperative set of workstations and routers with multicast capability. The MBONE has been in existence since 1992, primarily as a research and collaboration tool using multimedia applications. It has been greatly expanded from the original

Internet Engineering Task Force (IETF) video and audio multicasts, and now includes 24-hour world news audio sessions and NASA space missions, which use real-time audio and video transmissions.

With the 3Com implementation of IP multicasting, you can have the following advantages:

- Obtain audio and video transmissions using your existing infrastructure (over Ethernet, FDDI, or token ring) and on any media over which 3Com supports IP routing.
- Enable the development of entirely new classes of IP-based applications.
- Ease the migration of existing LAN-based multicast applications and distributed systems to an IP-based environment.
- Conserve bandwidth by reducing traffic and protect the host from receiving unwanted datagrams (only members of the group receive the multicast packet).
- Extend the benefits of multicast delivery beyond the confines of a single subnetwork as more multicast-capable IP routers are used.
- Access the MBONE across the Internet using tunneling.
- Experience complete compatibility with the UNIX program, mrouterd 3.5 and above (less compatibility with previous releases), the UNIX program implementing DVMRP that runs on most systems on the MBONE.

### **Multicast Addresses**

IP multicasting uses the destination address of the datagram to specify multicast delivery using Class D addresses in the range of 224.0.0.0 through 239.255.255.255.

The following Class D addresses are reserved:

- 224.0.0.0 – this address cannot be assigned to any group.
- 224.0.0.1 – this address is permanently assigned to the “all hosts” group, which includes all hosts and gateway participating in IP multicasting on a local network. No IP multicast address exists that refers to all hosts in the Internet.
- 224.0.0.2 – this address is assigned to all routers on a local network.
- 224.0.0.4 – this address is assigned to DVMRP routers on a local network.
- 224.0.0.5 – this address is assigned to all OSPF routers on a local network.
- 224.0.0.6 – this address is assigned to all OSPF designated routers and backup designated routers on a local network.
- 224.0.0.0 to 224.0.0.255 – these addresses are reserved for multicast applications that do not multicast more than one hop. Multicast packets addressed to these addresses are not forwarded outside the local network.
- 239.0.0.0 to 239.255.255.255 – these addresses are reserved for scoping purposes (a router is configured as a boundary router and multicast traffic does not cross the boundary) and for private multicast groups (traffic is not routed across the Internet).

IP multicast addresses can only be used as the destination address; they can never appear in the source address field of a datagram, nor can they appear in a source

route or record route option. For more information about IP addressing, see the Internet Addressing appendix.

### **Internet Group Management Protocol**

To participate in IP multicasting, multicast hosts and routers must have the IGMP operating. This protocol is the group membership protocol used by hosts to inform routers of the existence of members on their directly connected networks, and allows them to send and receive multicast datagrams.

Multicast routers learn about group membership when a host joining a new group sends an IGMP message to the group address declaring its membership. If the DVMRP Protocol is running, the local multicast router receives the group membership message and sends a DVMRP Graft message to its upstream router if it ever sent a DVMRP Prune message. If the MOSPF Protocol is running, the local multicast router receives the group membership message, establishes routes, and propagates the group membership information to other multicast routers throughout the internetwork.

Because membership is dynamic, local multicast routers periodically query hosts on the local network with Host Membership Query messages to determine which hosts remain members of which groups. These messages are periodically sent by the designated router (the one with the lowest IP address in DVMRP or the one with the highest router priority in MOSPF) to refresh their knowledge of membership present on a particular subnet. Hosts respond with Host Membership Report messages. If no host reports membership in a group after a query, the multicast router assumes that no host on the network remains in that group. If the DVMRP is running, the router sends a Prune message to its upstream router for the next data packet destined to this group and assumes that no other downstream routers are interested in this group. If the MOSPF Protocol is running, the router stops advertising group membership to other multicast routers. Hosts can also send Host Leaves Group messages whenever they want to leave a multicast group.

The information learned by the IGMP is stored in a local group membership database and is used by both the DVMRP and MOSPF Protocols.

### **Distance Vector Multicast Routing Protocol**

To propagate routing information among multicast routers, a multicast routing protocol such as DVMRP can be used. Multicast routers use the DVMRP to pass source subnet information among themselves, using the information to establish routes to deliver a copy of the multicast datagram to every subnet containing a member of the multicast group.

Like the RIP, the DVMRP passes information about known subnets and the cost to route between gateways. For each possible multicast group, the router imposes a routing tree on top of the graph of the physical interconnections. When a router receives a datagram destined for an IP multicast address, it sends a copy of the datagram over the network links that correspond to branches in the routing tree.

The 3Com implementation of the DVMRP applies the Reverse Path Multicasting (RPM) algorithm that allows for the shortest-path multicast tree to be pruned on demand. Pruning preserves bandwidth by removing multicast routers from the tree when no members for that group are on any directly connected subnets and no downstream routers are interested in that group (multicast packets do not

need to be received and are discarded by this router because no group members are attached).

The DVMRP uses a number of messages to discover neighboring routers. Some of these messages include the following:

- Probe – discovers neighbors that support multicast routing.
- Route Report – contains route information.
- Prune – destined to the parent router to detach it from the delivery tree if no members for that group are on any directly connected subnets.
- Graft – sent to an upstream router when a new member joins the group after a Prune message had previously been sent.
- Graft Acknowledge – sent to the downstream router to acknowledge the previous Graft message.

### Routing Table

Each DVMRP multicast router creates a routing table containing a list of routes learned from other multicast router's route report messages. Using these route report messages, the router builds a routing table and a shortest-path tree for each source.

The router also keeps track of the following links:

- Parent link  
A parent link is the expected interface to receive multicast packets from a source (the interface that leads to the previous-hop router back to the source).
- Child link  
For each (source, group) pair, the child links are the set of interfaces on which to forward multicast packets. The router uses the child link information to perform Reverse Path Broadcasting (RPB).
- Leaf link  
A leaf link is a child link that no router uses to reach a source. For a given source, if no members of a particular group on the subnet are associated with a leaf link, DVMRP truncates the leaf link from the shortest path tree using the Truncated Reverse Path Broadcasting (TRPB) algorithm.

The DVMRP router also assigns the following router functions:

- Designated router  
The router with the lowest IP address on a subnet becomes the designated router. The designated router is responsible for sending IGMP Host Membership Query datagrams on the subnet.  
When a multicast router starts, it considers itself to be the designated router until it receives a Host Membership Query or Report datagram from a neighbor router with a lower IP address.
- Dominant router  
To avoid duplicate multicast datagrams when more than one router exists on a virtual interface, one router is elected as the dominant router for a particular source. The dominant router is the router that is responsible for forwarding

multicast datagrams on a subnet for a source (it has a route to the source with the lowest metric on that virtual interface).

- Subordinate router

A subordinate router for a virtual interface is the downstream router that considers this interface to be its parent link. Information from a subordinate router helps the DVMRP router decide whether to truncate the shortest path tree. For each route entry, the subordinate router helps decide if the subnet for that virtual interface is a leaf subnet.

### Forwarding Table

In conjunction with the routing table, the DVMRP creates a forwarding table. The forwarding table contains group information (source and group pairs) that is applied to the routing table's shortest-path tree. The forwarding table helps the router forward multicast datagrams to each member of the group using the routing table's shortest-path tree.

The DVMRP router can receive Prune messages from downstream routers in the shortest-path tree if the attached subnet contains no group members for the particular (source, group) pair. In this way, the router can prune the shortest-path delivery tree, allowing datagrams to only be forwarded to the subnets in which the specified group is located. The DVMRP leaf router also prunes the shortest-path delivery tree if it no longer receives IGMP Host Membership Report messages or if all members have left a group. The forwarding table maintains an entry in its cache until the timeout period is reached. During the timeout period, if the DVMRP router learns that members have rejoined a group, it sends a Graft message to the upstream routers indicating that a member has rejoined and allows the branches of the shortest-path tree to reattach.

### Multicast Open Shortest Path First Protocol

To propagate routing information among multicast routers, a multicast routing protocol such as MOSPF can be used. MOSPF is an extension of the base version 2 OSPF Protocol and is backward compatible with OSPF (routers running OSPF interoperate with MOSPF routers). The introduction of multicast extensions does not impact unicast IP traffic. MOSPF routers identify other MOSPF-capable routers for forwarding multicast IP packets. Unlike DVMRP, where separate routing protocols for unicast and multicast packets are run, OSPF and MOSPF run a single copy of the protocol. But like DVMRP, MOSPF forwards multicast traffic based on both the source and destination address, known as source and destination routing.

The MOSPF Protocol does not provide the ability to tunnel through non-MOSPF capable routers. MOSPF routers must be directly interconnected with each other. Failure to do so may lead to nondelivery of multicast packets even though unicast connectivity is maintained.

While forwarding multicast packets, MOSPF may replicate packets along the way. The replication is performed only at tree branches where replication is absolutely necessary. Although multiple copies may be forwarded, the packet is not modified (except the TTL field, where it is decremented by 1 at each hop). No IP-over-IP encapsulation is performed. The destination address is always listed as Class D multicast address. To avoid packet duplicates, equal-cost multiple path forwarding in MOSPF is not possible.



When sending multicast IP packets, MOSPF conforms to link-layer encapsulation. Over Ethernet and FDDI interfaces, the mapping between IP multicast and datalink multicast address is used. Over other kinds of LAN interfaces, link-level multicast or broadcast is used. Over WAN media, IP multicast packets are encapsulated as unicast packets.

OSPF partitions the network topology into a number of routing domains, with ASBRs interconnecting routing domains. Within a routing domain, OSPF allows multiple areas to be interconnected by ABRs. Areas may be transit, stub, or backbone. MOSPF partitions the network topology in the same way as OSPF; the same topology for both OSPF and MOSPF can be used. For more information on the OSPF topology, see “Understanding IP Network Topology” in the Configuring IP Routing chapter.

### Learning Group Membership

MOSPF uses the IGMP Protocol to monitor multicast group membership on directly attached LANs. MOSPF periodically sends IGMP queries and listens to IGMP replies. The membership information learned is then used to build group-membership link state advertisements (LSAs).

On a LAN, only the designated MOSPF router (usually the one with the highest router priority) sends queries at the interval specified by the `-MIP QueryInterval` parameter to the “all hosts” address (224.0.0.1) and listens to IGMP replies. The MOSPF designated router (DR) processes IGMP replies and performs the IGMP maintenance work on the network. The DR is responsible for flooding group membership information throughout the routing domain by issuing group membership LSAs. When a new group is learned, MOSPF sends a new group membership LSA. When a group is aged out, MOSPF flushes the corresponding group membership LSA. When the MOSPF router resigns as the DR, it flushes all locally generated group membership LSAs.

In a mixed environment in which MOSPF and OSPF routers reside on the same LAN, an MOSPF router must become the DR to monitor group membership, generate group-membership LSAs, and forward multicast packets onto the LAN. Therefore, OSPF routers should be assigned a router priority of 0 to prevent them from becoming the DR, allowing an MOSPF router to become the DR.

### Shortest Path First Tree

MOSPF uses the group membership LSA with the OSPF database, which provides complete topology information about the area and routing domain. The group-membership LSAs describe the location and address of all multicast groups in an area and routing domain. The group membership database is built by the IGMP Protocol and enables delivery of multicast packets.

MOSPF routers use the group membership LSA information to compute the shortest path first (SPF) tree, which enables delivery of multicast packets to remote destinations. The SPF tree is rooted at the packet's source address toward all destination group members and describes the intermediate hops from the source to all possible destinations belonging to the same group. Different sources are likely to have different trees. The SPF tree is pruned only toward the intended destination; all paths and routers that do not lead to group members are pruned from the tree. A separate tree is built for each source and destination pair.

The SPF tree is computed on demand (when a packet is received). A cache entry is created with the source and destination pair; the upstream node and downstream interface information is recorded. The SPF tree is then discarded, freeing all resources along with it. The newly created cache entry is used for forwarding decisions, and the entry is stored in the forwarding database. Future received packets with the same source and destination pair can locate its forwarding decision from the database without resorting to another SPF computation.

### Forwarding Cache

Each MOSPF router in the path of a multicast packet makes its forwarding decision based on the contents of its forwarding cache. The forwarding cache is built from the local group database and the SPF tree. Each cache entry contains information about received multicast packets from the neighboring node (upstream router or LAN) and where multicast packets should be forwarded (downstream interfaces or MOSPF neighbors). Each downstream interface has a time-to-live (TTL) value associated with it. The TTL value indicates the number of hops a datagram can travel to reach the nearest multicast destination or be discarded. The hop count prevents packets from being uselessly forwarded and conserves bandwidth. The hop count is further restricted by the `-MIP THreshold` parameter.

The cached information is not aged or periodically refreshed; the information is kept as long as enough system resource are available, or until the next topology change. However, the forwarding cache may need to be flushed under the following circumstances:

- OSPF topology changes
- Group membership LSA changes with identical multicast destination
- Local group database changes with identical multicast destination

### Interarea Multicasting

When multicast routing occurs between areas (interarea multicasting), source and destination addresses may not reside in the same area, the ABR must have multiple copies of the OSPF link databases (one for each area), and the MOSPF router must build separate SPF trees for each area.

Recall that ABRs are responsible for interconnecting areas (transit or stub) to the backbone and other areas. The backbone area is considered a transit area, with area number 0 reserved for it. All ABRs must be connected to the backbone area, either directly or through virtual links. The ABRs are responsible for summarizing reachability information from the backbone to other areas, and from other areas to the backbone. These summaries take the form of summary LSAs.

When running the MOSPF Protocol, a portion of the OSPF ABRs must be configured through the `-MOSPF MABR` parameter as interarea multicast forwarders, which are ABRs with multicast extensions enabled. An interarea multicast forwarder must be an ABR, but not all ABRs need to be interarea multicast forwarder.

The interarea multicast forwarder calculates all the reachable group addresses from their areas. They convey group membership information to other areas by summarizing the group membership LSAs from their attached areas into the backbone. They do not summarize group membership information from the

backbone to other areas. All interarea multicast forwarders concurrently and independently perform this action.

After the router summarizes group membership LSAs into the backbone, the backbone area has complete information regarding all the reachable group memberships. The backbone area may not know the exact location of group members subnets (because that requires the detailed topology information from within the area), but it knows which area is interested in which group address. Nonbackbone areas have only group membership information for their area and do not know that some group members exist in other areas.

For multicast packets to flow between areas, all interarea multicast forwarders announce wild-card multicast receiver status (equivalent to the default route for unicast traffic) into attached areas. A wild-card multicast receiver is a router to which all multicast packets should be forwarded regardless of the multicast destination. With sufficient routing information in a backbone area, a wild-card multicast receiver is not needed. Interarea multicast forwarders do not announce wild-card multicast receiver into the backbone.

Wild-card multicast receiver status is automatic; no user configuration is required. In nonbackbone areas, all interarea multicast forwarders are wild-card multicast receivers. Backbone area do not need these receivers.

- When MOSPF routers are used between areas, they perform one SPF computation for the source and destination per attached area. Each area has its own link state database, and the SPF computation exclusively uses the LSAs within the area. The backbone area is treated the same as other areas.

### Interautonomous System Multicasting

When multicast routing occurs between autonomous systems (interautonomous system multicasting), some MOSPF routers must be configured as inter-AS multicast forwarders. These inter-AS multicast forwarders have additional routing information for forwarding multicast packets outside the routing domain. These inter-AS multicast forwarders can concurrently run another inter-AS multicast protocol in the same router or be configured with static external routes. However, the current implementation does not support static routes. Inter-AS multicast routers are configured through the `-MOSPF Dvmrp` and `PolicyControl` parameters.

The MOSPF Protocol guarantees that all inter-AS multicast forwarders receive all multicast packets. When multicast packets are received from outside the MOSPF domain, MOSPF assumes those packets reach the inter-AS multicast forwarder through a Reverse Path Forwarding algorithm. The DVMRP also uses a Reverse Path Forwarding algorithm.

All inter-AS multicast forwarders declare themselves as wild-card multicast receivers in the backbone area. After reaching the backbone area, all multicast packets are required to reach all inter-AS multicast forwarders regardless of destination.

---

## Multicast Routing Terms

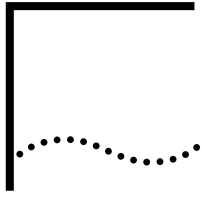
The following terms are used in this chapter to explain multicast routing:

|            |                                                              |
|------------|--------------------------------------------------------------|
| child link | The set of interfaces on which to forward multicast packets. |
|------------|--------------------------------------------------------------|

|                                                    |                                                                                                                                                                                                                                                                                                   |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Core-Based Trees (CBT)                             | With this protocol, all the group members share a multicast delivery tree. CBT builds a multicast tree based on a core router instead of the source. It builds one multicast tree per group instead of one per (source, group) pair.                                                              |
| designated router                                  | The router that is responsible for sending IGMP Host Membership Query datagrams. For DVMRP routers, the designated router is the router with the lowest IP address on the subnet. For MOSPF routers, the designated router is the router with the highest priority.                               |
| Distance Vector Multicast Routing Protocol (DVMRP) | This distance-vector protocol builds a shortest-path source-based delivery tree between each source (sender) and multicast group (receivers). It builds one multicast delivery tree per (source, group) pair. DVMRP has been implemented as a UNIX program on mrouterd routers for several years. |
| dominant router                                    | One of several routers on a link that is elected for a particular source. It is responsible for forwarding multicast datagrams on a subnet for a source.                                                                                                                                          |
| downstream router                                  | The router to which a multicast packet is forwarded.                                                                                                                                                                                                                                              |
| forwarding table                                   | A table containing group information (source and group pairs). This group information is applied to the routing table's shortest-path tree and helps the router forward multicast datagrams to each member of the group.                                                                          |
| leaf link                                          | A child link that no router uses to reach a source.                                                                                                                                                                                                                                               |
| multicast                                          | A technique that allows copies of a single packet to be passed to a selected subset of all possible destinations.                                                                                                                                                                                 |
| Multicast Open Shortest Path First (MOSPF)         | This protocol is an extension to OSPF that builds a shortest-path source-based delivery tree on demand.                                                                                                                                                                                           |
| parent link                                        | The expected interface that receives packets from a source; also the interface that leads to the previous-hop router back to the source.                                                                                                                                                          |
| Protocol Independent Multicast (PIM)               | This protocol is not an extension of any unicast routing protocol; it relies on the unicast routing protocols and is suited for large heterogeneous internetworks. It contains two modes: Dense and Sparse. Dense mode uses a similar algorithm to the one used by DVMRP.                         |
| Reverse Path Broadcasting (RPB)                    | A refinement of the RPF algorithm that eliminates duplicate broadcast packets.                                                                                                                                                                                                                    |

|                                            |                                                                                                                                                                                                                                                                                  |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reverse Path Forwarding (RPF)              | An algorithm that forwards multicast datagrams by computing the shortest (reverse) path tree from the source network to all possible recipients. The router forwards the packet further only if it considers the sending router as the next-hop address of the source multicast. |
| Reverse Path Multicasting (RPM)            | A refinement of the TRPB algorithm that provides on-demand pruning of the shortest-path multicast tree.                                                                                                                                                                          |
| routing table                              | A table containing a list of routes that are learned from other multicast router's route report messages.                                                                                                                                                                        |
| Truncated Reverse Path Broadcasting (TRPB) | A refinement of the RPB algorithm that only forwards packets to where they are wanted by pruning the shortest-path tree. This algorithm prunes branches of nonmember leaf networks.                                                                                              |
| upstream router                            | The router from which a multicast packet is received.                                                                                                                                                                                                                            |





# CONFIGURING THE VIRTUAL ROUTER REDUNDANCY PROTOCOL

This chapter describes how to configure the Virtual Router Redundancy Protocol (VRRP). VRRP allows a backup IP router to immediately take the place of a failed master router using the same IP and MAC addresses. A master router is a default router explicitly specified by end hosts.



*For conceptual information, see “How VRRP Works” later in this chapter.*

---

## Configuring VRRP

The procedures in this section describe how to configure VRRP.

### Supported Media and Protocols

VRRP supports the following media:

- Ethernet
- Fast Ethernet
- FDDI

You cannot run the following protocols on the same router as VRRP:

- CLNP, OSI
- APPN, LAA
- DECNET

VRRP can be used with IPv4 only.

### Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router using the procedure in the Configuring Basic Ports and Paths chapter.
- Configure IP routing on each bridge/router.

### Enable the Owner Router

The owner router is always the master router when it is active. Assign a virtual router identifier (VRID) to the owner router using:

```
ADD !<port> -VRRP VRid <vrid> <IP address> [, <IP address>]
```

<port> is the port on which you define VRRP. Specify the port that is connected to the LAN.

<vrid> is any number between 1 and 255 that you want to identify the VRRP owner IP addresses. If you have more than one owner router on a LAN, you must use unique VRIDs.

<IP address> is the IP address or addresses that you have assigned to the port in the IP service. You can specify up to five addresses. This IP address will be used by the backup router when the owner router goes down.

If you have multiple subnets on the port, you can add more than one VRID if you want a separate backup router for each subnet. You can add multiple VRIDs per router as long as each VRID is unique.

### Enable the Backup Routers

Assign a backup router to the owner router using:

```
ADD !<port> -VRRP BackUp [AsOwner] <vrid> <IP address> [,<ip address>]
```

Use this BackUp parameter on the backup router only. A bridge/router can be configured as both an owner router and a backup router, so long as each vrid is configured for only one function.

This parameter identifies one or more IP addresses, the associated vrid, and the port where they should be backed up. The owning router uses the VRID parameter, not the BackUp parameter.

[AsOwner] is an optional 3Com extension to VRRP that specifies that when this router takes over the IP addresses it will behave as though it owns the IP address; it will respond to PING, allow TCP connections, etceteras. This behavior is contrary to the RFC, but is useful for some applications, such as the DLSw resilient tunnel example application. (See the DLSw Resilient Tunnels section.)

Often in this type of application, there is no real owner of the IP address - it moves among the backup routers.

<port> is the port on which you define VRRP. Specify the port that is connected to the LAN. The VRRP port must be an Ethernet, Fast Ethernet, or FDDI interface.

<vrid> is the VRID defined on the owner router. The <vrid> value must match on the owner and all back up routers.

<IP address> is the owner router IP address or addresses. The address must match the owner router IP address exactly. The backup router's own IP address, configured in the IP service, must have the same subnet as the owner router.

### Setting Priorities for Multiple Backup Routers

When the master router goes down, it stops sending VRRP packets, enabling the backup router to become the master router.

If you have more than one backup router, which router becomes the master is determined by the following settings:

- Preempt — This option, set in the CONTROL parameter, allows a router that has a higher priority than the current master to assume control as the master router. NoPreempt reduces network instability caused by master router changes, so only use Preempt if you must ensure the primary backup router.
- HoldTime and skew time — The HOLDTime parameter sets the amount of time before the backup router declares the master router to be down. The total time consists of the hold time plus the skew time. The skew time depends on the priority of the router, and is either 500 ms or 1 second. The router with the lowest [hold time + skew time] will become master router first.



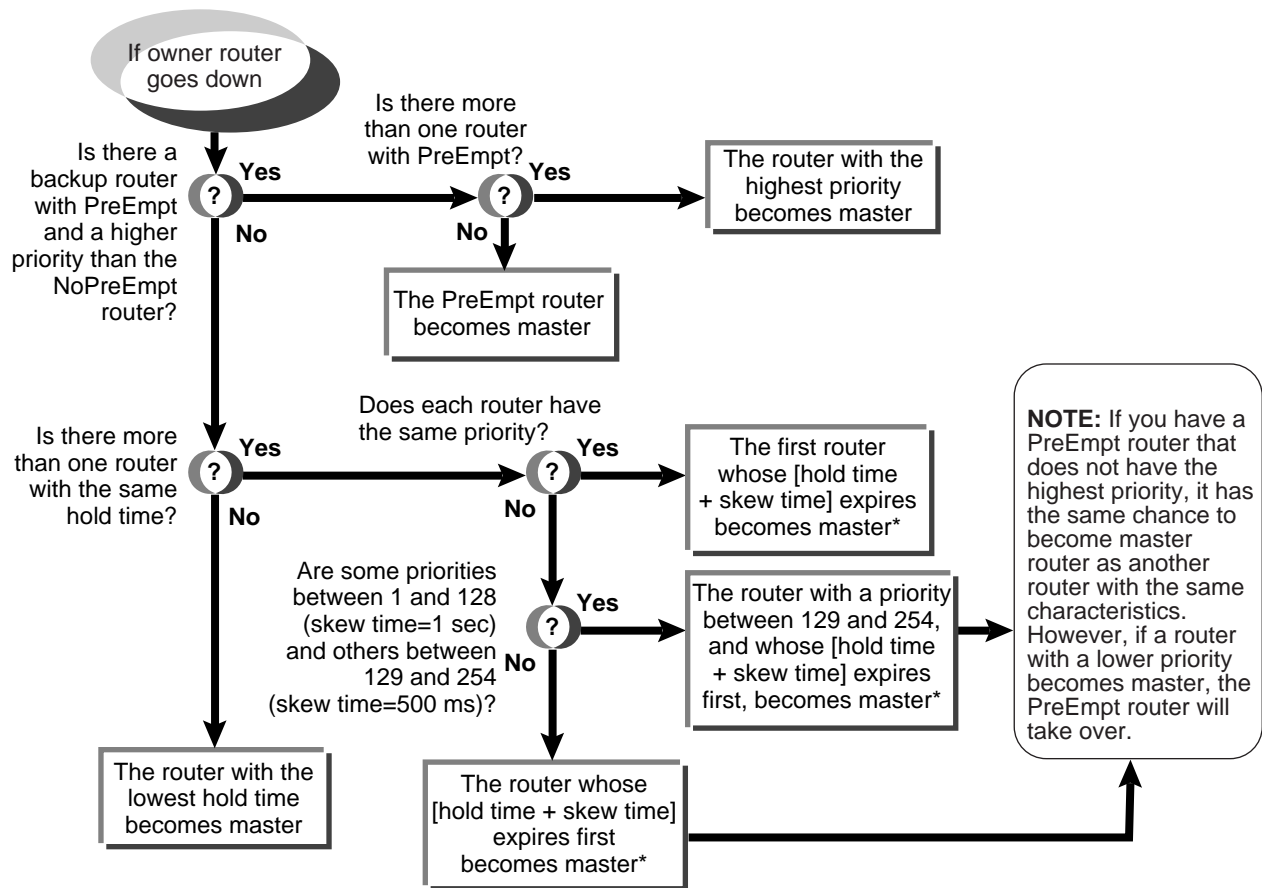
- PRIOrity** — If more than one router has the same PreEmpt setting, the router with the highest priority becomes master. Also, the priority determines the skew time, either 500 ms or 1 second.

The owner router is *always* master when it is active, because it is set for PreEmpt and has the highest priority (255).

If you have more than one backup router, it is recommended that you use the PRIOrity field to establish an order in which each backup router would become the master should the master router fail.

Figure 115 shows which router will become master when there are multiple backups.

**Figure 115** VRRP Priorities



\*You cannot control which router will become master if they have the same characteristics. Because of the timer on the NETBuilder bridge/router, the same [hold time + skew time] can vary by as much as 500 ms.

### Setting the Hold Time

The highest priority backup router should have the lowest hold time. Set the hold time for the highest priority router, then set the hold time to be one second longer for the next priority router. Add one second for each succeeding router, in order of priority.

Set the amount of time before the backup router declares the master router to be down using:

```
SETDefault !<port> -VRRP HOldTime = <hold time>(3 to 255 seconds) <vrid>
```

Where <hold time> is a number from 3 to 255 in seconds. The hold time should be at least three times the advertisement interval (see “Setting the Advertisement Interval” later in this chapter). The default is 3.

If you want a backup router to become the master no matter what the hold time is, you can specify PreEmpt in the CONTrol parameter and assign it the highest priority using the PRIOrity parameter.

### Setting the Priority

If you have more than one PreEmpt router, the PRIOrity determines which router becomes master. The priority is a value between 1 and 254.

The priority also determines the length of the skew time, which in conjunction with the hold time determines when the backup router can declare the master router down. If the priority is between 1 and 128, the skew time is 1 second. If the priority is between 129 and 254, the skew time is 500 ms. If the hold time is the same on two routers, but the priorities fall in different ranges, then the higher priority router will become master. If you have more than one router within a range, you cannot determine which router will become master. Because of the time on the NETBuilder bridge/router, the same [hold time + skew time] can vary as much as 500 ms. 3Com recommends setting the hold time itself at least one second apart to guarantee priority.

You can set the priority of a backup router to any number between 1 and 254 using:

```
SETDefault !<port> -VRRP PRIOrity = <priority number> <vrid>
```

### Enabling VRRP

Enable VRRP on each router using:

```
SETDefault !<port> -VRRP CONTrol = (Enable,[PreEmpt | NoPreEmpt]) <vrid>
```

PreEmpt allows a router that has a higher priority than the current master router to assume control as the master router. Because the owner router has the maximum priority, it is always the master router when it is active. You cannot set the owner router to NoPreEmpt. If you have more than one backup router, and you use NoPreEmpt (the default), the router will assume control only when the current master router fails, even if it has a higher priority. If you know that one backup router should always be the primary backup, you should specify PreEmpt in addition to giving it a high priority. If you do not have a strong preference, keep NoPreEmpt on all backup routers. NoPreEmpt reduces network instability caused by master router changes.

If the owner router goes down, and then comes up later, it will become the master router again without waiting for the backup to fail.

<vrid> is the VRID of the owner router.

### Ping/Telnet Virtual Router IP (VIP)

If the master of the virtual router is not the owner router, that is the owner router is down, Ping/Telnet Virtual Router IP (VIP) will get no response. This is the default

behavior as specified in the VRRP specification. By adding the "AsOwner" keyword to the VRRP BackUp command, the backup router will treat the IP address as if it were their owner. That is, it will respond to pings, and can be used as the destination of a TCP connection, for Telnet or DLSw, for example. This feature is demonstrated in the DLSw resilient tunnel scenario. (See the DLSw Resilient Tunnels section.)

### Disabling/Deleting VRRP

Before disabling/deleting a VRRP from the owner router, you must disable/delete the VRID from the backup routers. Failing to do so may cause the same IP address to be mapped to different MAC addresses on different nodes. This happens because the new master and the owner are both responding to the ARP request for the VIP. You may also see "Local-Col" displayed in response to a SHow -IP address command. The "Local-Col" will be cleared two hours after the problem is corrected (that is, after the vrid backup routers are disabled/deleted).



*The System IP address of a router should not be used as a VRRP Virtual Router IP (VIP). Doing so may cause two routers to respond to the same IP address. When a request (such as SNMP) is sent to the VIP, the response can be returned from either of the routers.*

---

## Customizing VRRP

This section describes how to customize your VRRP configuration.

### Setting the Advertisement Interval

Set the interval between VRRP packets sent by the master router using:

```
SETDefault !<port> -VRRP AdvertisementInt = <adv_time>(1 to 127 seconds)
<vrid>
```

Where <adv\_time> is a number from 1 to 127 in seconds. The default is 1 second. You must set the advertisement interval to the same value on all routers associated with a VRID.

---

## How VRRP Works

VRRP is a protocol between IP routers that allows backup routers to monitor the status of a master router. When the master router fails, the backup router can take over the function of the master router. The new master router keeps the IP and MAC address of the original master, so that hosts that are configured with a single default gateway do not have their network connectivity disrupted if the gateway fails.

With VRRP enabled, the master router sends out regular VRRP packets to indicate that it is alive. If the VRRP packets stop, the backup router adopts the IP and MAC address of the master, in addition to its own IP and MAC addresses. If you have more than one backup router, the router with the highest priority becomes the master router.

Each bridge/router running VRRP is either the owner or the backup router for a VRID. There can be only one owner for each VRID. The owner router owns the IP address, configured in the IP service, that is used by the backup router if the owner fails. The owner router is always the master router if it is active, because it has a higher priority than any backup router.

**MAC Address** The MAC address used by the master router is 00-00-5E-00-01-<vrid>. Each VRID has one MAC address associated with it, so a router that has multiple VRIDs will have a different MAC address for each VRRP interface.

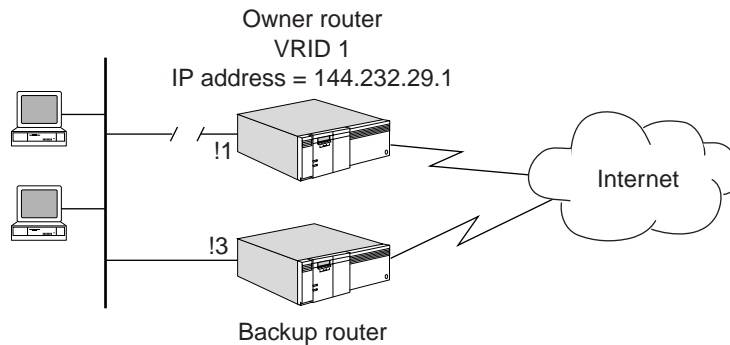
**Scenarios** This section includes the following scenarios:

- Gateway to a WAN
- Connecting Two LANs
- Load Sharing with Redundancy
- DLSw Resilient Tunnels

### Gateway to a WAN

Figure 116 shows a basic VRRP setup with one gateway router connecting a LAN to the Internet. If port 1 goes down, the backup router becomes master.

**Figure 116** Owner Router with One Backup Router



To configure VRRP on the network in Figure 116, follow these steps:

- 1 On the owner router:
  - a Create the VRID by entering:
 

```
ADD !1 -VRRP VRid 1 144.232.29.1
```
  - b Enable VRRP by entering:
 

```
SETDefault !1 -VRRP CONTrol = Enable 1
```
- 2 On the backup router:
  - a Back up the VRID 1 by entering:
 

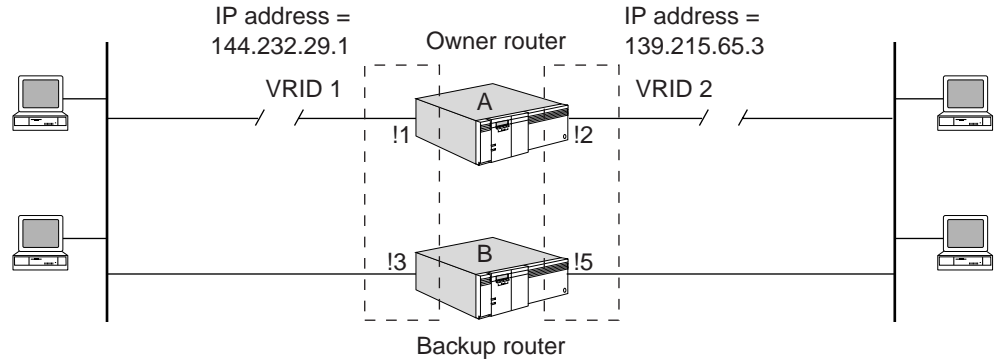
```
ADD !3 -VRRP BackUp 1 144.232.29.1
```
  - b Enable VRRP by entering:
 

```
SETDefault !3 -VRRP CONTrol = Enable 1
```

If the owner router fails, the backup router automatically takes over using the IP address 144.232.29.1 and the MAC address 00-00-5E-00-01-01.

### Connecting Two LANs

Figure 117 shows two LANs connected by a master router, with one backup router.

**Figure 117** Two VRIDs on One Router Connecting Two LANs

To configure VRRP on the network in Figure 117, follow these steps:

- 1 On router A:
  - a Add VRID 1 to port 1 by entering:
 

```
ADD !1 -VRRP VRid 1 144.232.29.1
```
  - b Add VRID 2 to port 2 by entering:
 

```
ADD !2 -VRRP VRid 2 139.215.65.3
```
  - c Enable VRRP on both ports by entering:
 

```
SETDefault !1 -VRRP CONTrol = Enable 1
 SETDefault !2 -VRRP CONTrol = Enable 2
```
- 2 On router B:
  - a Back up VRID 1 by entering:
 

```
ADD !3 -VRRP BackUp 1 144.232.29.1
```
  - b Back up VRID 2 by entering:
 

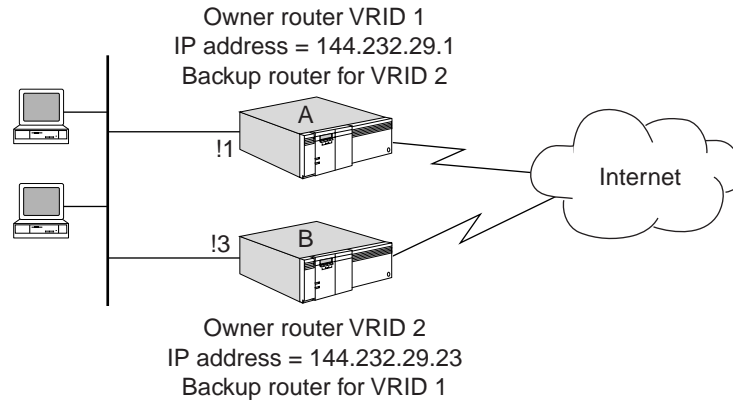
```
ADD !5 -VRRP BackUp 2 139.215.65.3
```
  - c Enable VRRP on both ports by entering:
 

```
SETDefault !3 -VRRP CONTrol = Enable 1
 SETDefault !5 -VRRP CONTrol = Enable 2
```

If the master router fails, the backup router automatically takes over. Port 2 uses IP address 144.232.29.1 and MAC address 00-00-5E-00-01-01. Port 5 uses IP address 139.215.65.3 and MAC address 00-00-5E-00-01-02.

### Load Sharing with Redundancy

Figure 118 shows two routers sharing the network traffic on subnet 144.232.29.x. Half of the hosts have router A configured as the default gateway and the other half have router B as the default gateway. Each router is configured to back up the other if one fails.

**Figure 118** Load Sharing with Redundancy

To configure VRRP on the network in Figure 118, follow these steps:

- 1 On router A:
  - a Add VRID 1 to port 1 by entering:
 

```
ADD !1 -VRRP VRid 1 144.232.29.1
```
  - b Back up VRID 2 by entering:
 

```
ADD !1 -VRRP BackUp 2 144.232.29.23
```
  - c Enable VRRP for each VRID by entering:
 

```
SETDefault !1 -VRRP CONTrol = Enable 1
SETDefault !1 -VRRP CONTrol = Enable 2
```
- 2 On router B:
  - a Add VRID 2 to port 3 by entering:
 

```
ADD !3 -VRRP VRid 2 144.232.29.23
```
  - b Back up VRID 1 by entering:
 

```
ADD !3 -VRRP BackUp 1 144.232.29.1
```
  - c Enable VRRP for each VRID by entering:
 

```
SETDefault !3 -VRRP CONTrol = Enable 2
SETDefault !3 -VRRP CONTrol = Enable 1
```

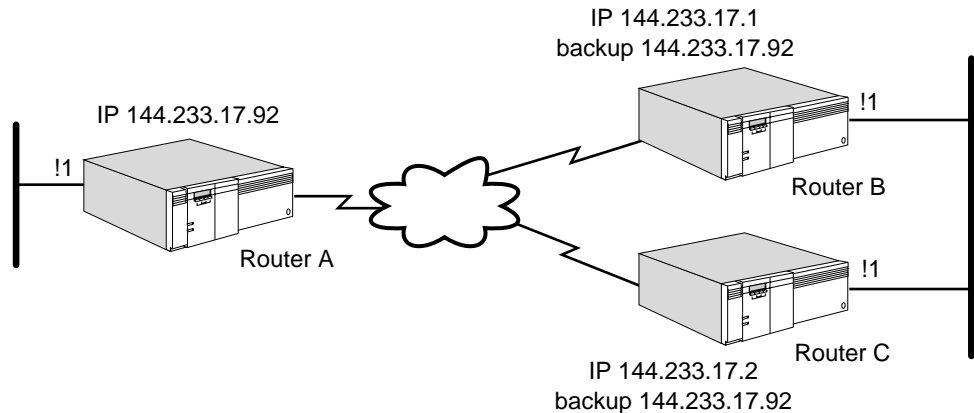
If router A fails, router B automatically takes over. Port 3 uses IP address 144.232.29.1 and MAC address 00-00-5E-00-01-01, in addition to its own IP and MAC addresses, 144.232.29.23 and 00-00-5E-00-01-02.

If router B fails, router A automatically takes over. Port 1 uses IP address 144.232.29.23 and MAC address 00-00-5E-00-01-02, in addition to its own IP and MAC addresses, 144.232.29.1 and 00-00-5E-00-01-01.

### DLSw Resilient Tunnels

Figure 119 shows a DLSw resilient tunnel. Router A is at a remote site, and is configured with a DLSw peer at 144.233.17.92. At the central site there are two routers: B, with IP address 144.233.17.1, and C with IP address 144.233.17.2. Both of these routers are VRRP “BackUp AsOwner” routers for the DLSw tunnel endpoint IP address 144.233.17.92. This configuration allows for DLSw redundancy. In addition, it prevents loops in an DLSw ethernet environment, because there is only one active tunnel at any time.

Figure 119 DLSw Resilient Tunnel



To configure VRRP on the network in Figure 119, follow these steps:

- 1 On router A:
  - a Enable LLC2 to receive LAN traffic on port 1 by entering:
 

```
SETDefault !1 -LLC2 CONTROL = Enable
```
  - b Configure a DLSw peer to the VRRP backup address by entering:
 

```
ADD !1 -DLSw PEER = 144.233.17.92
```
- 2 On routers B and C:
  - a Configure the routers to "Backup AsOwner" the DLSw tunnel endpoint on port 1 by entering:
 

```
ADD !1 -VRRP Backup AsOwner 1 144.233.17.92
```
  - b Enable VRRP for VRID 1 by entering:
 

```
SETDefault !1 -VRRP CONTROL = Enable 1
```
  - c Set the DLSw interface to be the tunnel endpoint address by entering:
 

```
SETDefault -DLSw Interface = 144.233.17.92
```
  - d Enable DLSw LLC2 traffic on port 1 by entering:
 

```
SETDefault !1 -LLC2 CONTROL = Enable
```

When routers A and B start, one of them "takes over" the IP address, and the other goes into a "hot standby" mode. Alternately, the VRRP PRIORITY and HoldTime could be set to be more deterministic.

For instance, in this example, if router A takes over the IP address and then fails, router B automatically takes over. Port 1 uses IP address 144.233.17.92, in addition to its own IP address 144.232.29.23. All DLSw TCP connections are reset by router B, because this router does not have any TCP connections. However, the remote routers automatically restart the TCP connections to router B. When router A reboots, it automatically goes into a "hot standby" mode to backup router B.



*In this example, router A was not made the real owner of the address, nor was VRRP Preempt set in the control parameter, because there was no reason to disrupt the DLSw connections and move them to router A.*

For more information about VRRP, point your browser at the following URL:

<http://ds.internic.net/internet-drafts/draft-ietf-vrrp-spec-06.txt>

## VRRP for Token Ring

When applying VRRP to Token Ring, special attention needs to be paid to source routing for multiple rings. (See the "RouteDiscovery" parameter in the SR Service Parameters chapter of *Reference for Enterprise OS Software*.) There are two approaches for running VRRP on Token Ring: binding the VRRP IP address to a functional address (functional address mode), or binding the VRRP IP address to a unicast address (unicast address mode).



*Functional address mode is the recommended mode. In unicast address mode, if source route is enabled on a VRRP router, then the same ring number must be assigned to all the ports with source route bridging enabled. In unicast mode, only two VRIDs are supported, where one is an owner and the other is a backup.*

### Functional Address Mode

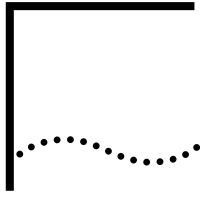
The functional address mode uses a functional address for the VRRP's VMAC address. This binds a VRRP IP address to a functional address. When an ARP request is for the VRRP IP address, the VRRP master sends an ARP response that binds the VRRP IP address to a functional address. Packets from the hosts that are addressed to the functional addresses are forwarded to the Spanning Tree explorer path. A new master listens to the functional address and forwards the packets destined to the functional address.

Since the functional address cannot be used as the source address, the real MAC address is used as the MAC source address in VRRP advertisements and the ARP responses. The binding between a VRRP IP (which has a unique VRID) and its functional address is static (preassigned by RFC[2338]).

### Unicast Address Mode

Similar to Ethernet and FDDI, Token Ring VRRP supports an unicast mode of operation. This mode binds a VRRP IP address to a unicast VRRP VMAC address. The unicast mode requires that the VRRP router to receive packets destined to different unicast MAC addresses. The MAC driver enters promiscuous mode, and the ARP request/reply packets contain the VRRP virtual MAC address as the source MAC address. Unicast mode requires all VRRP routers for the same VRID be connected to the same ring.





# CONFIGURING THE ROUTER DISCOVERY PROTOCOL

This chapter describes how to configure your system to use Internet Control Message Protocol (ICMP) Router Discovery Protocol (RDP) messages to dynamically generate a list of active neighbor router addresses. RDP is an extension of ICMP, and assists hosts in discovering neighboring routers. It is also a requirement per RFC 1812 for IP V4 routers on all connected networks that support either IP multicasting or IP broadcast addressing.

This chapter also describes how RDP works and gives guidelines for operating, managing, and troubleshooting it.



*For conceptual information, see "How RDP Works" later in this chapter.*

---

## Setting Up RDP

The procedure in this section describes the steps required to enable your system for RDP. Depending on your network requirements, you can use the default values of the parameters in the various services, or you can configure the router with custom values.

### Prerequisites

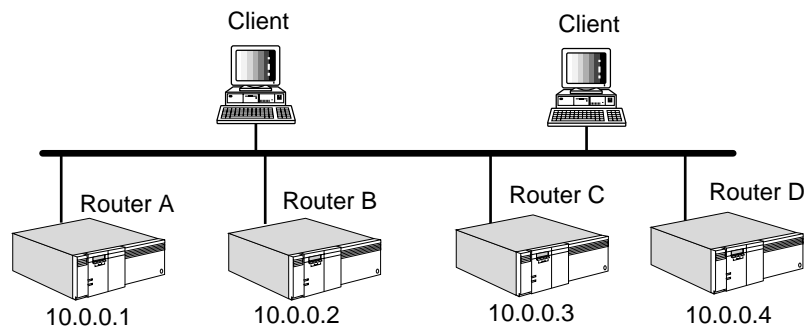
Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your ports and paths as described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure IP routing as described in the Configuring IP Routing chapter or IP multicast routing as described in the Configuring IP Multicast Routing chapter.

### Procedures

To set up RDP, see Figure 120 and perform these procedures.

**Figure 120** Routers Participating in RDP



## Defining Participating Routers

To create the list of routers that will participate in RDP, follow these steps:

- 1 Define the list of routers that will participate in the router discovery process using:

```
ADD RouterList <IP address>[NoAdvertise][<preference level>|Infinity]
```

Enter the IP address for each system in router mode to be advertised. The length of this list is limited only by available system memory. For example, to list the routers in Figure 120 for participation in RDP, enter:

```
ADD -RDP RouterList 10.0.0.1
ADD -RDP RouterList 10.0.0.2
ADD -RDP RouterList 10.0.0.3
ADD -RDP RouterList 10.0.0.4
```

- a You can also enter the IP address for each system in router mode, and assign a preference level to specify the default routers that will learn from router advertisement messages using:

```
ADD RouterList <IP address> <preference level>
```

The <preference level> option indicates the preferences for selecting the default router. For example, to designate a router as the default, enter:

```
ADD -RDP RouterList 10.0.0.1 100
```

The preference level option is a 32-bit, signed, two's-complement integer, which allows you to enter a definitive number to specify the selection criteria for the default router. The higher the value assigned, the more preference the router address has.

- b To indicate that an address *not* be used as a default router address, enter:

```
ADD -RDP RouterList 10.0.0.4 Infinity
```

The Infinity option indicates a minimum value (0x80000000) that prevents it from being picked up by hosts as a default router.

- c If you do not want a router to participate in the discovery process, enter its IP address with the NoAdvertise option. For example:

```
ADD -RDP RouterList 10.0.0.6 NoAdvertise
```

## Configuring the Timers

To configure the RDP timers, follow these steps:

- 1 Specify a value for the lifetime field in router advertisement messages using:

```
SETDefault !<port> -RDP LifeTime = <seconds>(4-9000) | Default
```

The default is 30 minutes (1800 seconds), and is valid only on routers in router mode.

- 2 Specify a value for the maximum interval between two router advertisement messages using:

```
SETDefault !<port> -RDP MMaxInterval = <seconds>(4-1800) | Default
```

The default is 10 minutes (600 seconds), which must be less than the value of the LifeTime parameter, and is valid only on routers in router mode.

- Specify a value for the minimum interval allowed between two router advertisement messages using:

```
SETDefault !<port> -RDP MInInterval = <seconds>(3-1800) | Default
```

The default is 75 percent of the MAXInterval value (nine minutes or 450 seconds), which must be less than the value set for MAXInterval, and is valid only on routers in router mode.

For example, the default lifetime of the router advertisement message is 30 minutes, and the default interval between the messages is ten minutes. To change these values to a lifetime value of 12 minutes and an interval of 6 minutes, enter:

```
SETD !1 -RDP LifeTime = 720
SETD !1 -RDP MAXInterval = 360
SETD !1 -RDP MInInterval = 300
```



*The value of MInInterval must always be less than that set for MAXInterval.*

### Enabling and Disabling RDP

To enable or disable RDP, follow these steps:

- Enable or disable RDP globally using:

```
SETDefault !<port> -RDP CONTrol = ([Auto | Enable | Disable],
[Multicast | Broadcast])
```

The default is Auto, which enables RDP on local area networks, but not wide area networks.

- To enable RDP on a wide area network, enter:

```
SETDefault !1 -RDP Control = Enable
```

- To specify that packets are multicasted, enter:

```
SETDefault !1 -RDP Control = Multicast
```

When the system is set in host mode, the Multicast option sends router solicitation messages out with the IP destination set to the all-routers address (244.0.0.2).

When the router is set in router mode, the Multicast option sends router advertisement messages out with the IP destination address set to the all-host IP address (244.0.0.1). This is the default mode.

- To specify IP broadcasting when it is enabled, enter:

```
SETDefault !1 -RDP Control = Broadcast
```

Both router solicitation and router advertisement messages are sent out with the IP destination set to the limited-broadcast IP address (255.255.255.255).

### Discovering Neighboring RDP Routers

To discover neighboring RDP routers by having the system send out router solicitations, use:

```
DiscRouteRs [!<port> | <source IP>] [Broadcast] [<timeout (1-30
seconds)>]
```

For systems in host mode (!0), to discover neighboring routers, specify either an outgoing port number or one of the system source IP addresses to be sent with the router solicitations. After the command is entered, the system transmits router solicitations every second until reaching the time set with the timeout option. During the timeout period, any router advertisements that are received are displayed.

---

## Verifying the RDP Configuration

To verify that RDP is recognized by the IP network, display the values associated with RDP using:

```
SHow [!<port>] -RDP CONFIguration
```

---

## Troubleshooting the RDP Configuration

You can troubleshoot the RDP operation using one or more of these steps:

- 1 Display the set of interfaces enabled or disabled for RDP using:

```
SHow [!<port>] -RDP CONTrol
```

- 2 Display the router list by entering:

```
SHow -RDP RouterList
```

- 3 Delete one or all of the interfaces enabled or disabled for RDP using:

```
DElete -RDP RouterList {<IP address>|ALL}
```

- 4 Flush the router list and allow the routes to be relearned by entering:

```
FLush -RDP RouterList
```

This command only works when the bridge/router is in host mode.

- 5 Display the value of the router advertisement lifetime field using:

```
SHow [!<port>] -RDP LifeTime
```

- 6 Display the value of the maximum interval between router advertisement messages using:

```
SHow [!<port>] -RDP MAxInterval
```

- 7 Display the value of the minimum interval between router advertisement messages using:

```
SHow [!<port>] -RDP MInInterval
```

---

## How RDP Works

RDP is a process defined by RFC 1256 that allows a router to use two messages, router advertisements and router solicitations, to discover the addresses of neighboring routers. The RDP process works only on routers enabled for the process that are in the same subnetwork. The router listens for router advertisements, or can solicit an address by sending a router solicitation message.

The discovery process is dynamic because the list collected contains only the addresses of active routers. Routers that are not actively sending router advertisements are dropped from the list, and are reinserted in the list only when they come back up and begin sending messages again.

RDP is not a routing protocol; it only allows hosts to keep track of neighboring routers, not which router is best to reach a particular destination. This protocol uses two ICMP messages to provide a simple router discovery method that provides a list of router addresses on a multicast link without manual configuration and that is independent of the routing protocol being used.

Although RDP cannot make routing decisions, if a host makes a poor choice for a first-hop router for a destination, it receives an ICMP Redirect message identifying a better route.

**RDP Features** The router advertisement message includes a preference level for each advertised router address. When a host system must choose a default router, it is expected to select router addresses with a high preference level. You set this preference level when you specify the list of routers that will participate in the RDP process.

To make sure that hosts ignore routers that go down, the router advertisement message also includes a lifetime field that specifies the maximum length of time that advertised addresses are to be considered valid by hosts. The default advertising rate is every ten minutes, and the default lifetime span is 30 minutes. The defaults minimize the load imposed on the links by the periodic transmission of the messages, but you can change the defaults as needed. For example, you may want to decrease the lifetime value so that you can become aware of routers that go down before the 30-minute period is up.

**Other Timer Considerations** When an interface enabled for advertising the router address becomes active, the router begins transmission of periodic router advertisement messages. The interval between the first three messages cannot be greater than 16 seconds. After these first three messages, however, the interval is randomly chosen from the values configured for each interface. A new random interval is chosen for each transmission to reduce the possibility that all routers will transmit packets at the same time.

Periodic advertisements are either multicasted to the all-host address (244.0.0.1) or broadcasted to the limited-broadcast address (255.255.255.255), depending on the system configuration. The router also transmits advertisements in response to host solicitations. When the source IP address is not set to 0, the reply is unicasted to the host; otherwise, it is multicasted to the configured address.

The host transmits no more than three RDP router solicitation messages when an interface becomes active. The first message is transmitted within one second and then retransmitted at 3-second intervals. Transmissions stop when a host receives a valid router advertisement message. Router solicitations can be configured to be either multicast to the all-routers address (244.0.0.2) or broadcast to the limited-broadcast address (255.255.255.255).

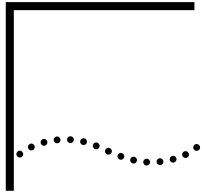
---

## RDP Terms

The following terms are used in this chapter to explain RDP:

|                |                                                                                                                                                                                                                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| multicast link | A link over which the IP multicast or IP broadcast services are supported, and can include media such as satellite, point-to-point, and store-and-forward networks such as SMDS.                                                                                                                             |
| neighbor       | Router with an IP address belonging to the same subnet.                                                                                                                                                                                                                                                      |
| default router | The router address that has the highest preference level. Unless a host has been redirected or is configured to use a specific router address, it must choose a default router address for a particular destination. You can set the preference level to encourage or discourage use of a particular router. |





# CONFIGURING THE REMOTE POLLING PROTOCOL

This chapter describes how to configure the Remote Polling Protocol (REMP). REMP monitors the reachability of target network devices and collects network performance data for a list of configured remote targets by periodically polling the targets for an echo response.

---

## Configuring REMP

When a target cannot be reached, you can be notified by an SNMP trap (if configured) or REMP.

You can use REMP anytime you want to monitor the reachability of any target from the NETBuilder bridge/router.

Targets can be added to the remote-polling list in one of two ways: statically or dynamically.

## Adding Static Targets

Static targets are manually configured using the UI, SNMP, or the LoadConfig command. Static targets are saved to the remote polling configuration file, and are preserved across reboots. This group of targets is always polled.

In this example, a static target is added with an IP address of 129.213.10.1 and default settings. A static target with IP address of 129.213.20.2 is added with specific settings (For example; ATtempts=8, Mode=RetryOnFailure, Wait=20 seconds).

To configure a static target using the REMP Service, follow these steps.

- 1 To set this example up with the UI, enter:  

```
ADD -REMPolling RemoteTarget 129.213.10.1
ADD -REMPolling RemoteTarget 129.213.20.2 A 8 M rof W 20
```
- 2 To set this up with SNMP for a target with an IP address of 129.213.10.1:
  - a Send an SNMP "createRequest" command to create a remPollEntry for 129.213.10.1.
  - b Send an SNMP "valid" command to set the entry just created for 129.213.10.1 to valid.

For a target with IP 129.213.20.2

- a Send an SNMP "createRequest" command to create a remPollEntry for 129.213.20.2.
- b Send SNMP set request to set the following objects for 129.213.20.2:  

```
remPollMode to RetryOnFailure,
remPollAttempts to 8,
```

`remPollTimeout` to 2000

- c Send an SNMP "valid" command to set the entry just created for 129.213.20.2 to valid.

### Dynamic Targets

Dynamic targets are added to the REMP Service automatically. Dynamic targets are not saved to the REMP configuration files, and they are not preserved across reboots.

There are three dynamic target groups:

- Remote Authentication Service (RAS) — RAS group targets are obtained when a RAS session is established. Each RAS entry is removed from the polling list when the RAS session is torn down. The RAS option is available on software packages that support RAS, only.
- Virtual Leased Line (VLL) — VLL group targets are obtained when you configure a VLL port using the following command:

```
ADD -l2t VLeasedLine <Dest_IP_Address> {Protocol = <PPTP
| 12TP>}
```

VLL targets are removed from the polling list when the VLL configuration is deleted.

- Tunnel Peer (TP) — TP group targets are obtained when you configure a port using one of the following commands:

- `ADD <!port> -Port DNL "ip address"`
- `ADD <!port> -Port VirtualPort IPIP remote_ipaddr`
- `ADD <!port> -Port VirtualPort IPIP p2mp`, followed by  
`ADD -ip addr <ip1> IPIP <ip2>`

TP targets are removed from the polling list when the port configuration is deleted.

---

### Target Group Priority

REMP support a maximum of 100 targets in the polling list. Because of this, a priority is used to determine which target(s) are added to the list when the list is full.

By default, the priority order is as follows:

- Static group
- RAS group
- TP group
- VLL group

For example, if the polling list is at capacity with 55 static targets, 15 RAS targets, and 30 VLL targets, and you want to add an additional static target, one of the VLL targets is deleted from the list to make room for the new static target.

In the case when a lower priority target needs to be deleted to make room for a higher priority target, a target in the lowest priority group that was added first (first-in-first-out) is removed.



More than one group can be assigned the same priority. If this case, the FIFO deletion rule is also applied.

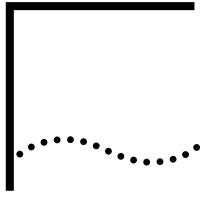
**Configuring Priority**

To configure the group priority, use the following command:

```
SETDefault -REMPolling PRIOrity = <priority_level> (1-4) <Static | RAS |
TP | VLL>
```

The REMP Polling Service creates a log file. You can configure the log file to be either ASCII or binary format. The AuditLog Service controls the distribution of messages to the Syslog server(s).





# CONFIGURING UDP BROADCAST HELPER

This chapter describes the User Datagram Protocol (UDP) Broadcast Helper feature. This feature allows applications in the Transmission Control Protocol/Internet Protocol (TCP/IP) stack to forward broadcast packets through a gateway (router) and to another network segment. The broadcast packets are typically requests from clients for access to servers, which may contain address, configuration, or name information.

A common application for UDP Broadcast Helper is related to the Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP). UDP Broadcast Helper for BOOTP and DHCP assists clients with finding their boot servers when the boot servers are located through a router and on another network segment.

3Com implements the BOOTP and DHCP relay agents in the bridge/router software, allowing existing BOOTP clients to interoperate with DHCP servers. The clients and associated servers do not need to reside on the same IP network or subnet, and changes to the client's initialization software is unnecessary.

This chapter provides information on how to configure UDP Broadcast Helper through the UDPHELP Service and how to verify the configuration. It also provides information on how to configure and customize the configuration.



*For conceptual information, see "How UDP Broadcast Helper Works" later in this chapter.*

---

## Configuring UDP Broadcast Helper

UDP Broadcast Helper allows you to configure up to 32 UDP ports on your bridge/router using the `ADD -UDPHELP ActivePorts` command.

UDP Broadcast Helper supports several names of well-known services. The names of these services are mapped to specific UDP port numbers. (The name-to-UDP port mappings are also referred to as *built-in names*.) You can configure UDP ports using built-in names. Table 26 lists the supported service names, the UDP port numbers they are mapped to, and the mnemonic name for each name-to-UDP port mapping.

**Table 26** Supported Service Name-to-UDP Port Mappings

| UDP Port Description | UDP Port Number (Decimal) | Mnemonic Name |
|----------------------|---------------------------|---------------|
| Daytime              | 13                        | DAYTIME       |
| Time                 | 37                        | TIME          |
| Host name server     | 42                        | IEN116        |
| Domain name server   | 53                        | DNS           |

**Table 26** Supported Service Name-to-UDP Port Mappings (continued)

| UDP Port Description          | UDP Port Number (Decimal) | Mnemonic Name |
|-------------------------------|---------------------------|---------------|
| (continued)                   |                           |               |
| TACACS – database service     | 65                        | TACACS        |
| Bootstrap protocol server     | 67*                       | BPSERVER      |
| Trivial file transfer         | 69                        | TFTP          |
| HOSTS2 name server            | 81                        | HOSTS2        |
| NIC host name server          | 101                       | NIC           |
| Simple file transfer protocol | 115                       | SFTP          |
| NetBIOS name service          | 137                       | NBNAME        |
| NetBIOS datagram service      | 138                       | NBDATA        |
| AppleTalk Name Binding        | 202                       | ATNBP         |
| AppleTalk zone information    | 206                       | ATZIS         |

\* BOOTP and DHCP use the same UDP port numbers: server port (67 decimal) and client port (68 decimal).

The UDP ports and built-in name mappings listed in Table 26 are reserved and cannot be changed or reconfigured.

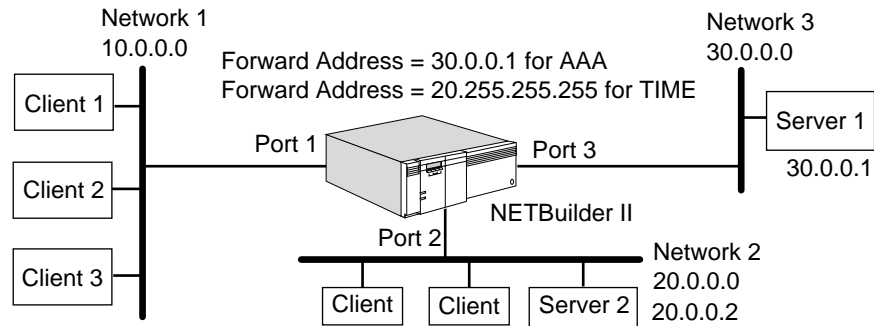
**Prerequisites**

Before beginning this procedure, complete the following tasks:

- Log on to the bridge/router with Network Manager privilege.
- Set up ports and paths according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Set up the client and server LANs and WANs according to the Configuring IP Routing chapter.
- Examine your network configuration and determine upon which bridge/routers UDP Broadcast Helper should be enabled.
- To determine what services are available through UDP Broadcast Helper, see Table 26. The number of services you want to configure determines the number of UDP ports you must configure.
- For each UDP port you intend to use, determine which networks or servers should receive related broadcast packets.
- Determine the IP addresses of the networks and servers that should receive broadcast packets.

**Procedure**

To set up UDP Broadcast Helper, see Figure 121 and follow these steps:

**Figure 121** Configuring UDP Broadcast Helper

A UDP port is part of an entity address and not related to an interface (port) on the bridge/router. In the command syntax, the UDP port does not need to be preceded by an exclamation point (!).

- 1 Enable UDP Broadcast Helper by entering:

```
SETDefault -UDPHELP CONTROL = Enable
```

- 2 Determine which UDP ports your bridge/router will be listening to or helping. Add each of these UDP ports to an active ports list using:

```
ADD -UDPHELP ActivePorts {<UDP port> | <name>}
```

You can specify a UDP port by either UDP port number or name. If you specify a UDP port by name, the name can be either a built-in or a name that you define.



If you want to specify a UDP port by a defined name, you must map the name to a UDP port number first as described in step 3, then add the UDP port to the active ports list as described in this step. To specify a UDP port by a defined name, you must perform step 3 first.

For example, to add UDP port 100, enter:

```
ADD -UDPHELP ActivePorts 100
```

To add a UDP port with the built-in name TIME, enter:

```
ADD -UDPHELP ActivePorts TIME
```

TIME is the name of a service that has a UDP port number mapped to it (see Table 26). In addition to specifying this UDP port by its built-in name, you can also specify this UDP by the port number mapped to this service. For example, you can enter:

```
ADD -UDPHELP ActivePorts 37
```

To add a UDP port with a name you define, for example, AAA, enter:

```
ADD -UDPHELP ActivePorts AAA
```

- 3 If you added a UDP port and specified it by port number, you can optionally define a name for the port and map the name to the port number. If you added a UDP port and specified it by a built-in name, skip this step and go on to step 4. If you want to add a UDP port and specify it by a name you defined, you must map the name to a UDP port number.

Use:

```
ADD -UDPHELP Name <name string> <UDP port>
```

For example, to map the defined name AAA to UDP port number 100, enter:

**ADD -UDPHELP Name AAA 100**

- 4 For each UDP port you added to the active ports list, 3Com recommends that you set up a list of networks and servers that should receive UDP broadcast packets.

After you add a UDP port to the active ports list, the bridge/router automatically forwards broadcast packets destined for the UDP port to all interfaces. You do not need to set up a list of networks and servers that should receive UDP broadcast packets. However, 3Com strongly recommends limiting the networks and server that receive UDP broadcast packets to help prevent broadcast storms and loops.

You can use one of the following syntaxes:

```
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address> <subnet mask>
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address> <subnet mask>
 [Ones | Zeroes]
ADD -UDPHELP ForwardAddress <UDP port or name> <list of interfaces>
```

You can add up to 32 addresses to the forward address list.

For example, using the network configuration shown in Figure 121, add server 1 to a list for UDP port AAA by entering:

**ADD -UDPHELP ForwardAddress AAA 30.0.0.1**

The bridge/router forwards broadcast packets destined for UDP port AAA to server 1 only.

To add network 2 to a list for UDP port TIME, enter:

**ADD -UDPHELP ForwardAddress TIME 20.0.0.0**

The bridge/router forwards broadcast packets destined for UDP port TIME to all nodes on network 2.



*The bridge/router does not rebroadcast packets through X.25, Frame Relay, and SMDS interfaces. You must add the IP address of each server to the list of servers that must receive UDP broadcast packets.*

- 5 To limit the reach of a broadcast packet and the potential duration of broadcast storms, 3Com recommends you specify the default number of seconds that pass before a broadcast packet is discarded. Use:

```
SETDefault -UDPHELP TTLOverride = <seconds>(1-255)
```

Upon receiving a client's request packet, the bridge/router assigns the packet a time-to-live (TTL) value. The bridge/router assigns the lowest TTL value among the following possible sources:

- The TTL value of the incoming request packet minus one
- The TTL value configured by the -UDPHELP TTLOverride parameter
- The TTL value configured by the -IP DefaultTTL parameter

If the TTL value configured by the -UDPHELP TTLOverride parameter is the lowest, the bridge/router forwards the packet with the TTL value configured by this parameter, which overrides the other TTL values.

For more information on the UDPHELP Service parameters used in this procedure, see the UDPHELP Service Parameters chapter in *Reference for Enterprise OS Software*. For more information on the -IP DefaultTTL parameter, see the IP Service Parameters chapter in *Reference for Enterprise OS Software*.

## Relaying BOOTP and DHCP Traffic

UDP Broadcast Helper allows you to set up BOOTP and DHCP so clients can boot from an unspecified server, which may be located through a router and on another network segment. The bridge/router forwards the BOOTPREQUEST packet and DHCP messages from a booting client to a server that can respond with the client's IP address.

If your network is quickly growing or changing, you may want to use the UDP Broadcast Helper for BOOTP instead of configuring a client to boot from one particular server, and then have to reconfigure the client to boot from another server if the network configuration changes.

By supporting both the BOOTP and DHCP relay agents, the bridge/router software allows existing BOOTP clients to interoperate with DHCP servers. BOOTP and DHCP clients and their associated servers often times do not reside on the same IP network or subnetwork. If the bridge/router software does not provide support for a relay agent, every subnet that has BOOTP and DHCP clients is required to have a BOOTP and DHCP server.

### Prerequisites

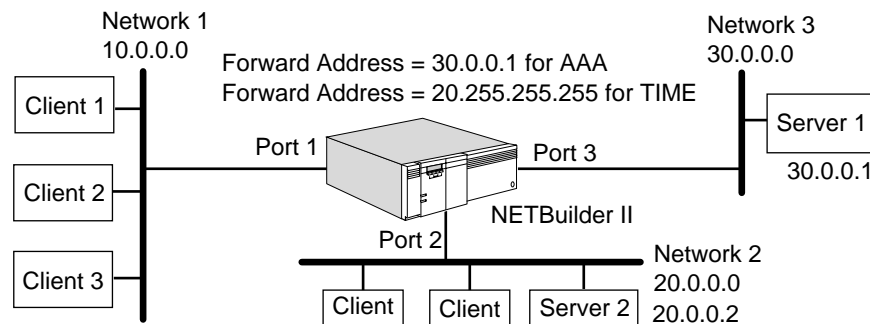
Before beginning this procedure, complete the following tasks:

- Log on to the bridge/router with Network Manager privilege.
- Set up ports and paths according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Set up the client and server LANs and WANs according to the Configuring IP Routing chapter.
- Examine your network configuration and determine which bridge/routers UDP Broadcast Helper for BOOTP should be enabled upon.
- Determine which networks or servers should receive BOOTPREQUEST packets.
- If possible, determine the IP addresses of the networks or servers that should receive BOOTPREQUEST packets.

### Procedure

To configure UDP Broadcast Helper for BOOTP and DHCP, see Figure 122 and follow these steps:

**Figure 122** Configuring UDP Broadcast Helper for BOOTP



- 1 Enable UDP Broadcast Helper by entering:

```
SETDefault -UDPHelp Control = Enable
```

- 2 Add a UDP port for the BOOTP or DHCP server to the active ports list.

You can specify either the built-in name BPSERVER or the UDP port number 67, which is mapped to built-in name BPSERVER. Both BOOTP and DHCP use the same UDP port numbers.

Enter either:

```
ADD -UDPHELP ActivePorts bpserver
```

or

```
ADD -UDPHELP ActivePorts 67
```

- 3 For UDP port 67 or BPSERVER, 3Com recommends that you set up a list of networks and servers that should receive the BOOTPREQUEST broadcast packets.



*If your bridge/router is configured to boot from a server that is accessed through an X.25, Frame Relay, or SMDS interface, you must perform this step. The bridge/router does not rebroadcast BOOTPREQUEST packets over X.25, Frame Relay, or SMDS interfaces.*

For an SMDS network, the group address functions as a LAN broadcast.

For X.25 and Frame Relay networks, the router duplicates the packet and forwards it to each configured or dynamically learned neighbor.

You need to configure the ForwardAddress parameter to eliminate unnecessary LAN broadcast packets using one of the following syntaxes:

```
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address> <subnet mask>
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address> <subnet mask>
 [Ones | Zeroes]
ADD -UDPHELP ForwardAddress <UDP port or name> <list of interfaces>
```

You can add up to 32 addresses to the forward address list.

If you know the specific IP address of the server (or the network IP address where the servers resides) from which the client should obtain its IP address, add the address to the list.

For example, if the address of the server that responds to the BOOTPREQUEST packets is 10.1.0.1, you can add this address to the list by entering:

```
ADD -UDPHELP ForwardAddress 67 10.1.0.1
```

In the next two examples, you can specify the mnemonic name BPSERVER instead of 67.

To forward BOOTPREQUEST packets to all servers on a specific network, enter:

```
ADD -UDPHELP ForwardAddress 67 10.0.0.0
```

The bridge/router stores address 10.255.255.255 in the list, meaning that all servers (hosts) on network 10 will receive the BOOTPREQUEST packet.

- 4 Optionally, configure the bridge/router to detect unauthorized BOOTP and DHCP servers using:

```
ADD -UDPHELP AuthDHCPserver <IP address>
```

Specify the addresses of authorized servers. You can add up to 32 servers to the list.

Any BOOTPREPLY or DHCP OFFER packet received with an IP source address that does not match any server's IP address on the list is discarded, a system message is entered, and an SNMP trap is sent. For information about the trap, see "AuthDHCPserver" in *Reference for Enterprise OS Software*.



For more information on the parameters used in this procedure, see the UDPHELP Service Parameters chapter in *Reference for Enterprise OS Software*.

This completes the basic configuration for UDP Broadcast Helper for BOOTP and DHCP. Information on customizing the configuration of UDP Broadcast Helper for BOOTP is described later in this chapter.

## Verifying the Configuration

This section summarizes the commands you need to know to verify UDP Broadcast Helper (including UDP Broadcast Helper for BOOTP) configuration and obtain related statistics.

### Checking Parameter Settings

You can check the settings of all parameters associated with UDP Broadcast Helper and UDP Broadcast Helper for BOOTP by entering:

```
SHOW -UDPHELP CONFIGURATION
```

### Getting Statistics

You can obtain statistics related to UDP Broadcast Helper and BOOTP by entering:

```
SHOW -SYS STATISTICS -UDPHELP
```

Statistics for UDP Broadcast Helper are displayed. For information on the elements of the display, see the Statistics Displays appendix.

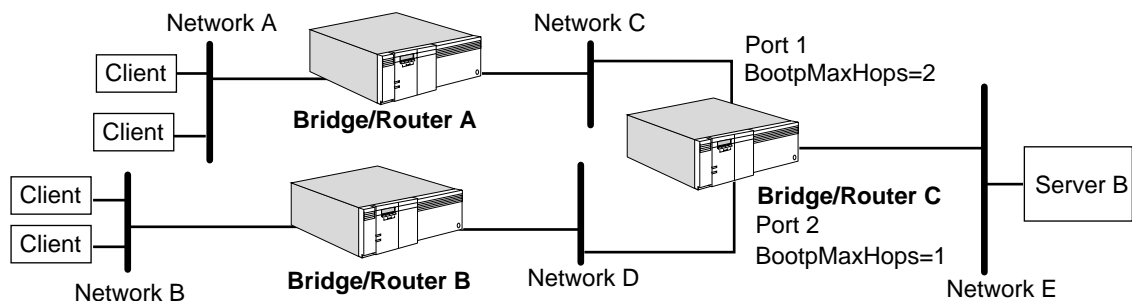
## Customizing the Configuration for BOOTP

You can customize UDP Broadcast Helper for BOOTP configuration by configuring the `BootpMaxHops` and `BootpThreshold` parameters in the UDPHELP Service. The `BootpMaxHops` parameter limits the number of hops that a BOOTPREQUEST packet can make on a network. The `BootpThreshold` parameter prioritizes and forwards BOOTPREQUEST packets to a server according to a predetermined plan and determines which clients are booted first.

### Limiting the Number of Hops

By configuring the `BootpMaxHops` parameter and limiting the number of hops, you can control how far a BOOTPREQUEST packet can travel on a network. For example, if your network configuration is similar to that shown in Figure 123, you can set the `BootpMaxHops` value on bridge/router C so that clients in a given area of the network can only boot from a specific server or servers.

**Figure 123** Limiting the Number of Hops for BOOTPREQUEST Packets



### Prerequisites

Before beginning the procedure, make sure that you have configured UDP Broadcast Helper for BOOTP as described earlier in this chapter.

## Procedure

For the following procedure, assume that a client on Network A needs to send BOOTPREQUEST packets to server B on network E. Because you do not know the IP address of server B and you have not configured the ForwardAddress parameter on any of the bridge/routers, each bridge/router will continue to forward the packet out each of its ports and flood the network with packets. To control this flood of packets, you can configure the BootpMaxHops parameter as follows:

- 1 On port 1 of bridge/router C, configure the BootpMaxHops parameter to 2 by entering:

```
SETDefault !1 -UDPHELP BootpMaxHops = 2
```

- 2 On port 2 of bridge/router C, configure the BootpMaxHops parameter to 1 by entering:

```
SETDefault !2 -UDPHELP BootpMaxHops = 1
```

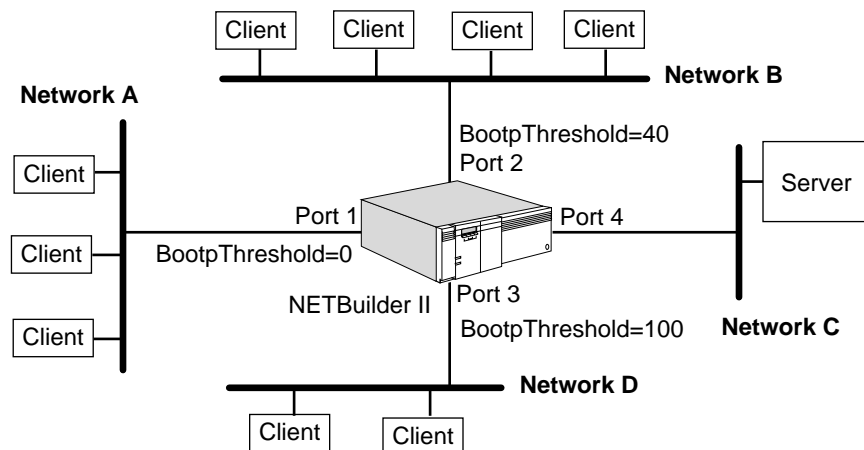
When bridge/router C receives BOOTREQUEST packets from the clients on network A, it forwards the packets to the server on network E. However, bridge/router C receives and discards the BOOTPREQUEST packets from the clients on network B because the BootpMaxHops parameter value is set to 1 on port 2. Bridge/Router C discards the BOOTPREQUEST packets because the packets have already traversed one gateway, which is bridge/router B.

For additional information on the BootpMaxHops parameter, see the UDPHELP Service Parameters chapter in *Reference for Enterprise OS Software*.

## Determining Order of Booting

By configuring the BootpThreshold parameter in the UDPHELP Service, you can determine which clients are booted first. For example, if your network configuration is similar to that shown in Figure 124, you can set the BootpThreshold value on each bridge/router port so that clients are booted according to a predetermined plan.

**Figure 124** Determining Which Clients are Booted First



## Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure UDP Broadcast Helper for BOOTP as described earlier in this chapter.
- Determine which clients you want to boot first.

## Procedure

For the following procedure, assume that according to your predetermined plan, you want the clients on network A to be booted first, then the clients on network B, and then the clients on network D. You need to set the `BootpThreshold` parameter on bridge/router ports 1, 2, and 3 to different values so that the bridge/router will prioritize and forward the `BOOTPREQUEST` packets to the server in the proper order. To determine which clients are booted first, follow these steps:

- 1 Set the `BootpThreshold` value on port 1 to the lowest value of all three ports.

To change the setting, enter:

```
SETDefault !1 -UDPHELP BootpThreshold = 0
```

- 2 Set the `BootpThreshold` value on port 2 to the next lowest value of all three ports by entering:

```
SETDefault !2 -UDPHELP BootpThreshold = 40
```

- 3 Set the `BootpThreshold` value on port 3 to a value greater than that set for ports 1 and 2 by entering:

```
SETDefault !3 -UDPHELP BootpThreshold = 100
```

When all the clients send out `BOOTPREQUEST` packets (the `Seconds Elapsed` Field in the `BOOTPREQUEST` packet is initially set to 0) at the same time, the bridge/router forwards the packets received on port 1 because the `Seconds Elapsed` Field and `BootpThreshold` value match. The bridge/router discards the packets received on port 2 and 3 because the `Seconds Elapsed` Field in these packets is less than the `BootpThreshold` value configured for ports 2 and 3.

The clients on networks B and D increase the `Seconds Elapsed` Field value in the `BOOTPREQUEST` packets and resend the packets. When the `Seconds Elapsed` Field value is greater than or equal to the `BootpThreshold` value on port 2, the bridge/router forwards the packets from the clients on network B to the server on network C. The bridge/router continues to discard the `BOOTPREQUEST` packets from network D until the `Seconds Elapsed` Field value is greater than or equal to the `BootpThreshold` value for port 3.

For additional information on the `BootpThreshold` parameter, see the `UDPHELP Service Parameters` chapter in *Reference for Enterprise OS Software*.

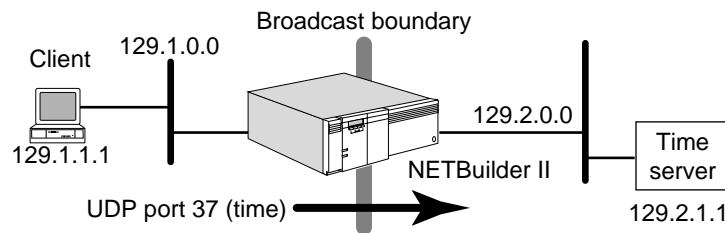
---

## How UDP Broadcast Helper Works

When boot servers are located through a router on another network, UDP Broadcast Helper helps BOOTP and DHCP clients to locate the server and retrieve address, configuration, and name information. Without the implementation of this feature, broadcast packets will not be propagated outside of the same network.

Broadcast packets generally do not traverse a router; however, there are some situations in which it is useful to propagate broadcast packets to other networks.

For example, in the topology shown in Figure 125, a client on network 129.1.0.0 may require access to a time server on network 129.2.0.0. Normally broadcast requests from the client on network 129.1.0.0 would not be forwarded to servers on network 129.2.0.0; however, you can configure UDP Broadcast Helper to allow the forwarding of broadcast requests to servers on network 129.2.0.0.

**Figure 125** Sample UDP Broadcast Helper Topology

UDP applications are identified within a packet by “well-known” port numbers. You can configure the bridge/router to allow broadcast packets to well-known port 37, which is the port number mapped to built-in name TIME for the time service, through to network 129.2.0.0.

### BOOTP and DHCP Protocols

The BOOTP Protocol is built on the client-server model and allows a single BOOTP reply to specify many items needed for a client to boot, including the client IP address, the address of a gateway, and the address of a server.

The DHCP Protocol is an extension of the BOOTP Protocol and is also built on the client-server model. DHCP is specifically designed for servers in large network environments that have nomadic users and complex TCP/IP software configurations.

DHCP not only allows a host to automatically allocate reusable IP addresses and additional configuration parameters needed for client operations, it also allows the client/server host to configure host parameters not directly related to the IP Protocol. This feature allows the host to exchange packets with any other host on the Internet. However, DHCP does *not* register newly configured hosts with the Domain Name System and is *not* used to configure routers.

The 3Com implementation of UDP Broadcast Helper feature includes the BOOTP and DHCP relay agent, which allows clients and their associated servers not residing on the same IP network or subnetwork to communicate. Without the relay agent, every subnet that has BOOTP and DHCP clients would be required to have a BOOTP and DHCP server.

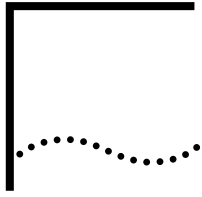
Although the BOOTP and DHCP Protocols uses the same UDP port numbers (67 and 68), they have some important differences as follows:

- DHCP allows IP addresses to be “leased” for a fixed length of time.
 

Groups of hosts that do not need permanent IP addresses can lease an address from a limited pool of addresses. Also, a host that is only temporarily connected to the network can be assigned an IP address because the addresses can be reused when they are no longer needed by the original host.
- DHCP packet length is longer than BOOTP.
 

The additional packet length allows a DHCP server to provide the client with all the IP configuration parameters that it needs to operate.
- DHCP is a more complicated protocol than BOOTP.
 

DHCP has seven message types; BOOTP uses only two. In addition, DHCP requires complex state machines.



# BUILDING INTERNET FIREWALLS

This chapter describes how to configure an Internet firewall on a NETBuilder II bridge/router and a model 227 SuperStack II NETBuilder bridge/router. This chapter provides a conceptual overview of a firewall and gives guidelines for operating and managing it successfully.



*For conceptual information, see "How a Firewall Works" later in this chapter.*

---

## Setting Up an Internet Firewall

The procedure in this section describes how to configure an Internet firewall. To configure the NETBuilder II bridge/router and SuperStack II bridge/router to perform firewall functions, you must set parameters in the FireWall Service.

Figure 126 shows two levels of firewall protection set up using the NETBuilder II bridge/router and the SuperStack II bridge/router. The first firewall is the model 227 SuperStack II bridge/router, which connects the Internet to a server subnet. The server subnet is where most Internet servers, such as the mail server and the WWW server, are located. The firewall on the SuperStack II bridge/router is enabled on port 3.

The second firewall is the NETBuilder II bridge/router, which connects the server subnet to the internal corporate network. The firewall on the NETBuilder II bridge/router is enabled on port 7. Traffic can flow freely between the other interfaces on the NETBuilder II bridge/router because the firewall will not be enabled on those interfaces. This permits the NETBuilder II bridge/router to not only perform firewall functions but also perform high-speed routing for internal networks.

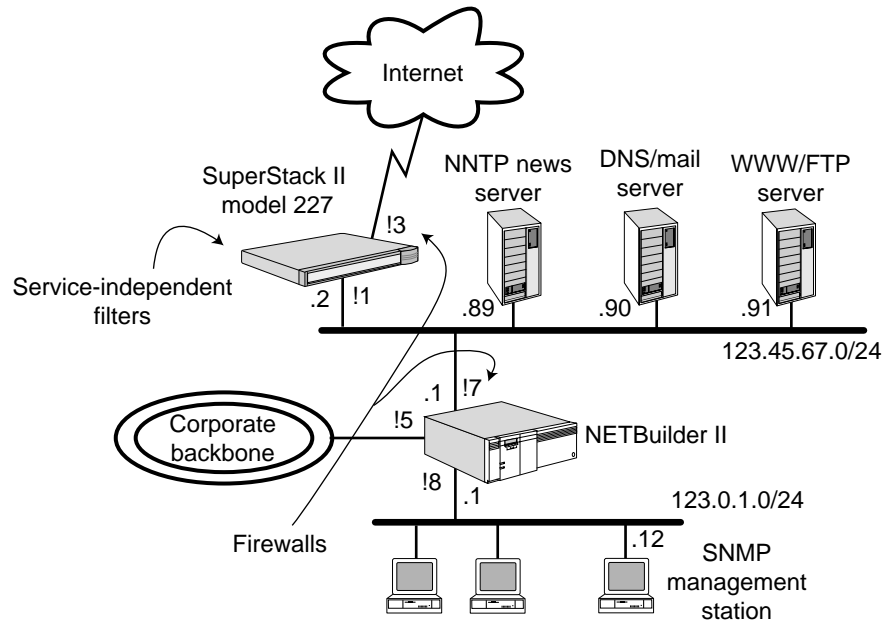
Although the NETBuilder II bridge/router is a secondary firewall, it is really the primary defense for internal networks since the internal servers are directly reachable from the Internet and they have a much higher chance of being compromised. Access from these internal servers to internal networks should be limited, and the servers should be configured using secured applications.

### Prerequisites

Before beginning this procedure, complete the following tasks:

- Read the information beginning with "How a Firewall Works" through "Setting Up System Logs" later in this chapter.
- Log on to the system with Network Manager privilege.
- Configure your ports and paths as described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Figure 126 NETBuilder Bridge/Router Firewall Example



### Defining Your Firewall Stance

You can choose between two firewall stances: “Everything not specifically permitted is denied” or “Everything not specifically denied is permitted.” The stance assumed in this chapter is “Everything not specifically permitted is denied.”

To define the basic stance of your firewall and decide whether log messages will or will not be recorded, follow these steps:

- 1 On the SuperStack II bridge/router, use:

```
SETDefault !<port> -FireWall DefActionIn = ([Permit | Deny], [Log | NoLog])
```

and

```
SETDefault !<port> -FireWall DefActionOut = ([Permit | Deny], [Log | NoLog])
```

- 2 On the NETBuilder II bridge/router, use:

```
SETDefault !<port> -FireWall DefActionIn = ([Permit | Deny], [Log | NoLog])
```

and

```
SETDefault !<port> -FireWall DefActionIn = ([Permit | Deny], [Log | NoLog])
```

The deny stance means that after all of the filters have been applied to a packet, and no actions have been taken, the packet must be dropped. You can explicitly deny specific types of traffic within your rules, which would stop traffic that you find dangerous or unnecessary before the system has to check that traffic against all of the other rules.

For more information, see the DefAction and Log parameters in the Firewall Service Parameters chapter in *Reference for Enterprise OS Software*.

## Continuing Routing Functions

Even while operating as a firewall, the NETBuilder II bridge/router and SuperStack II bridge/router must continue to perform the functions they were originally designed to perform, that is, they must continue to execute routing protocols so that they can correctly forward packets.

If your bridge/router is running OSPF or RIP, see the sections for those parameters the FireWall Service Parameters chapter in *Reference for Enterprise OS Software*.

Assuming OSPF is the routing protocol in use, to allow OSPF packets to come in and go out on all LAN interfaces, follow these steps:

- 1 On the SuperStack II bridge/router, enter:

```
ADD !3 -FireWall OSPF Permit
```

- 2 On the NETBuilder II bridge/router, enter:

```
ADD !7 -FireWall OSPF Permit
```

If RIP is being used, the syntax is the same. If BGP-4 or EGP is being used, you need to write a generic filter to allow any of these protocols to work properly. See "Generic Filters".

## Configuring OAM Procedures

Your operations, administration, and maintenance (OAM) procedures must keep working. Examples of OAM procedures include Telnet, Internet Control Message Protocol (ICMP) (Ping), File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), and Simple Network Management Protocol (SNMP). For information on how to configure a firewall for these services, see the TelnetIn and TelnetOut, FTPIn and FTPOut, TFTP, ICMP, and SNMP parameters in the FireWall Service Parameters chapter in *Reference for Enterprise OS Software*.

In the following procedures, 123.0.1.0/24 is the IP address of the management subnet, and 123.0.1.12/32 is the IP address of the SNMP management station. The syntax "a.b.c.d/x" denotes "the address a.b.c.d with the top x bits significant for comparison."



*The management station does not need to be directly attached to one of the firewall bridge/routers. The management subnet could be anywhere on the corporate network. For the purpose of this procedure, we assume only that there is such a network with a common IP prefix/mask.*

All of the filters used in the following steps are designed to control traffic to and from the bridge/routers. At this stage, no attempt is made to control traffic through the bridge/routers; that procedure is covered in "Blocking Unwanted Traffic".

### Configuring Telnet

To configure Telnet on both bridge/routers, complete this step:

- To allow users on the management LAN to Telnet into the SuperStack II bridge/router, on the NETBuilder II bridge/router enter:

```
ADD !7 -FireWall TELnetOut Permit From 123.0.1.0/24 to 123.45.67.2
```

This command guarantees that the NETBuilder II bridge/router will not block traffic from management stations to the SuperStack II bridge/router. No blocking actions are taken on Telnet traffic to the NETBuilder II bridge/router itself, because the firewall is not enabled on port 8.

## Configuring TFTP

NETBuilder II and SuperStack II bridge/routers sometimes use TFTP to perform file transfers or network booting. The bridge/router always functions as a TFTP client, while the management station functions as a TFTP server.

To configure TFTP on both bridge/routers, complete this step:

- To accept TFTP packets from the SuperStack II bridge/router to the management station, on the NETBuilder II bridge/router enter:

```
ADD !7 -FireWall TFTP Permit From 123.45.67.2 To 123.0.1.0/24
```

This command allows both TFTP request packets (123.45.67.2 -> 123.0.1.0/24) and TFTP response packets (123.0.1.0/24 -> 123.45.67.2). The NETBuilder II bridge/router will have no problem accessing the management server because the firewall is not enabled on port 8.

## Configuring ICMP (Ping)

To configure ICMP on both bridge/routers, allow the SuperStack II bridge/router to send and receive ICMP (Ping) message. On the NETBuilder II bridge/router, enter:

```
ADD !7 -FireWall ICMP Permit
```

## Configuring SNMP

SNMP (another UDP-based protocol) has two sides. On the receive side, there are management requests. On the transmit side, there are two kinds of packets: responses to management requests and traps. The following two filters allow the SuperStack II bridge/router to send SNMP traffic to the SNMP management station and receive traffic from there. 3Com is explicitly denying SNMP that originates elsewhere on the Internet.

To allow the SuperStack II bridge/router to be SNMP manageable, on the NETBuilder II bridge/router, enter:

```
ADD !7 -FireWall SNMP Permit From 123.45.67.1 To 123.0.1.12/32
```

```
ADD !7 -FireWall SNMP Permit From 123.0.1.12/32 To 123.45.67.1
```

## Configuring FTP

FTP can be used to move files between any bridge/router and an FTP server.



*If you do not use the FTP feature on your bridge/routers, you can skip this section.*

The SuperStack II bridge/router needs to be able to FTP to and from the management subnet.

On the NETBuilder II bridge/router, enter:

```
ADD !7 -FireWall FTPIn Permit From 123.45.67.1 To 123.0.1.0/24
```

## Verifying the Configuration

After the OAM filters have been defined, turn on IP Firewall and verify the configuration of both bridge/routers by following these steps:

- 1 Turn on the firewall on the SuperStack II bridge/router and the NETBuilder II bridge/router respectively by entering:

```
SETDefault !3 -FireWall CONTROL = Filter
```

```
SETDefault !7 -FireWall CONTROL = Filter
```

- 2 Verify current functionality on both bridge/routers.





```

Filters (3rd Priority)-----Bytes-Active@-----idle--Permit-Deny-Log
InFilter good 34 Mar25 12:21
OutFilter 0 -
DefAction (4th priority)-----Permit-Deny-Log
Deny 0 0 0

```

**Blocking Unwanted Traffic** To block unwanted traffic on the SuperStack II bridge/router and NETBuilder II bridge/router, follow these steps:

- 1 Configure external protection for the SuperStack II bridge/router.



*TCP-based services such as TELnet, FTP, SMTP, NNTP, DNS, Gopher, and Archie are much safer than non-TCP-based services. Avoid using non TCP-based services on your Internet connections. Due to the CPU requirements of DenySrcSpoofing, 3Com recommends that it be turned on only where absolutely required. Turn on DenySrcSpoofing on interfaces receiving external traffic to repel source spoofing attacks. Internal interfaces may not require such checking.*

The filters in this section apply to traffic through the bridge/router, not to and from traffic as described earlier.

- a Block hacker tricks.

These are nonintrusive filters that only adversely affect traffic from those individuals who are trying to break into your site. All normal traffic will proceed as usual.

On the SuperStack II bridge/router enter:

```
SETDefault !3 -Firewall CONTROL = DenySrcSpoofing
```

On the NETBuilder II bridge/router enter:

```
SETDefault !1 -FireWall CONTROL = DenySrcSpoofing
```

- b Allow remote secondary name servers to talk to your external name server.

These commands allow TCP-based domain name service (DNS) server-to-server traffic to be sent between 123.45.67.90 (the IP address of the name server in this example) and several remote name servers (for example, two off-site secondary name servers). Multiple name servers can be accommodated by adding more of these commands.

To configure the SuperStack II bridge/router, use:

```
ADD !1 -Firewall DNSSvrSvr Permit From 123.45.67.90/32 To <IPaddr>
```

```
ADD !1 -Firewall DNSSvrSvr Permit From 123.45.67.90/32 To <IPaddr>
```

- 2 Allow mail to be sent from the Internet to an external mail host on both bridge/routers.

SMTP is a particularly vulnerable service on many UNIX workstations because users run it mostly everywhere, and common vendor-supplied versions have many security holes in their implementations (not referring to the protocol here). Because of this vulnerability (and the difficulty of keeping all of the internal machines up-to-date with the latest version of "sendmail"), 3Com only allows hosts on the Internet to establish SMTP connections to the external mail host.

To configure the SuperStack II bridge/router, enter:

```
ADD !3 -FireWall SMTPOut Permit From 123.45.67.90/32
```

```
ADD !3 -FireWall SMTPIn Permit To 123.45.67.90/32
```

The NETBuilder II bridge/router must be configured to allow SMTP connections from anywhere on the corporate network (123.0.0.0/8 in this example) to anywhere on the Internet. You can also force all internal mail to go through the external mail host, but that is not usually a requirement. Allow SMTP to come in from the corporate backbone. On the other LAN interfaces, with "special treatment" for the perimeter network, only accept SMTP traffic in from the main mail host, and only to machines within the corporate address (123.0.0.0/8). On all other interfaces, SMTP is allowed to go out as long as it is coming from within the corporate address.

To configure the NETBuilder II bridge/router, enter:

```
ADD !7 -Firewall SMTPIn Permit From 123.45.67.90/32 To 123.0.0.0/8
ADD !7 -Firewall SMTPOut Permit From 123.0.0.0/8
```

- 3 Configure Hypertext Transfer Protocol (HTTP) and World Wide Web (WWW) connections on both bridge/routers.

Allow remote HTTP connections to and from the WWW server only; allow internal WWW browsers to go out.

On the SuperStack II bridge/router, allow HTTP traffic from the Internet to the external WWW server only. Block traffic to the rest of the corporate network.

To configure the SuperStack II bridge/router, enter:

```
ADD !3 -Firewall HTTPIn Permit To 123.45.67.91/32
ADD !3 -Firewall HTTPOut Permit
```

The first filter allows the Internet to access your WWW server; the second filter allows your internal users to originate HTTP traffic to anywhere on the Internet.

To configure the NETBuilder II bridge/router, enter:

```
ADD !7 -Firewall HTTPOut Permit
```

This command explicitly permits HTTP out through the NETBuilder II bridge/router. If this step is not taken, the DefaultAction parameter blocks all HTTP traffic in and out on all ports.

- 4 Allow remote news feeds to get to the external news server.



*This step is optional.*

Allow the internal Network News Transfer Protocol (NNTP) server to connect over the SuperStack II bridge/router LAN interface in the outgoing direction by entering:

```
ADD !3 -Firewall NNTPOut Permit From 123.45.67.89/32
```

Allow the external NNTP servers to make connections through the SuperStack II bridge/router Internet link, but only to the external NNTP server using:

```
ADD !3 -Firewall NNTPIn Permit From <IPaddr> To 123.45.67.89/32
ADD !3 -Firewall NNTPIn Permit From <IPaddr> To 123.45.67.89/32
```

The <IPaddr> variable in the first command is the IP address of the first external news feeder. The <IPaddr> variable in the second command is the IP address of the second news feeder. If you have 10 external news feeds, then you will need 10 filter rules like the first two.

Allow NNTP traffic to cross the NETBuilder II bridge/router by entering:

```
ADD !7 -Firewall NNTPIn Permit
ADD !7 -Firewall NNTPOut Permit
```

If NNTP is not explicitly permitted using these rules, the DefaultAction parameter settings deny it.

---

## Configuring a Firewall for IP Security Protocol Encrypted Packets

When using a firewall and you configure the IP Security Protocol on your IP router, additional configuration is required to allow the encrypted packets to pass through the firewall.

When you configure the IP Security Protocol, you add an IPSEC policy to your configuration. This policy consists of an action, the packets types that require the action, and the source and destination addresses between which the action occurs. Encapsulated Security Payload (ESP) and Authentication Header (AH) IP security packet headers may be specified.

### Specifying Packet Handling

When the IP Security Protocol is configured for ESP or AH packets, you need to specify how packets with ESP or AH headers are handled by the firewall.

To specify ESP packet handling through the firewall, use:

```
Add !<port> -FireWall IPsecESP permit | deny [Log [0-7]] [From <IPaddr/mask>] [To <IPaddr/mask>] [NextHop <IPaddr>]
```

For example, to allow ESP packets through the firewall, enter:

```
Add !1 -FW IPsecESP permit 10.1.1.1 to 20.2.2.2
```

To specify AH packet handling through the firewall, use:

```
Add !<port> -FireWall IPsecAH permit | deny [Log [0-7]] [From <IPaddr/mask>] [To <IPaddr/mask>] [NextHop <IPaddr>]
```

For example, to allow AH packets through the firewall, enter:

```
Add !1 -FW IPsecAH permit 10.1.1.1 to 20.2.2.2
```

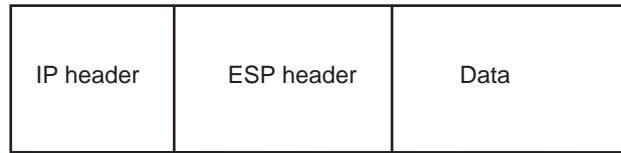
### Support for IP Security

IP Security supports the following two header types:

- Authentication header (AH)
- Encapsulating security payload (ESP) header

AH and ESP headers are used to provide encryption and the authentication mechanism between the two communicating entities. The primary difference between the authentication provided by ESP and AH is the extent of coverage. A detailed description of these headers is beyond the scope of this document.

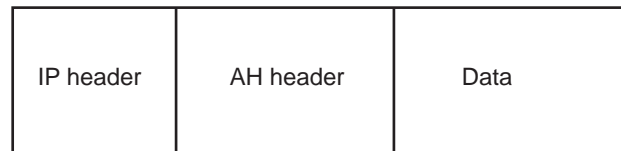
**ESP Header** In transport mode, the ESP header is inserted after the IP header and before the upper layer protocol header, for example, TCP, UDP, and ICMP. In tunnel mode, the ESP header is inserted after the IP header and before an encapsulated IP header. A typical IP packet with ESP header is shown in Figure 127.

**Figure 127** ESP Header

If an ESP header is present in an IP packet, the protocol field in the IP header will contain the value 50 (protocol id for the ESP).

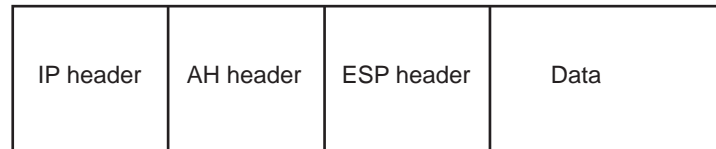
**AH Header** In transport mode, the AH header is inserted after the IP header and before the upper layer protocol header, TCP, UDP, or ICMP or before any other IP security header that have been inserted, for example ESP. In tunnel mode, the AH header is inserted after the IP header and before an encapsulated IP header.

A typical IP packet with AH header in it is shown in Figure 128.

**Figure 128** AH Header

If an AH header is present in an IP packet, the protocol field in the IP header will contain the value 51 (protocol id for the AH).

A typical IP packet with AH and ESP headers is shown in Figure 129

**Figure 129** AH and ESP Header

## User-defined Services

Several predefined services are provided, but you can also create your own service definition. For example, HTTP can be allowed on several unofficial but well known TCP ports 8080, 8000, 8001, and 8888. After you create this service definition, you can see the service name.

### Defining a Service

Being able to define a service allows you to reference a service name when you define filter rules.

When defining a service, use:

```
ADD -FireWall UserDefService <service name> <protocol> [Src <compare>
<port | port1-port2>] [Dst <compare> <port | port1-port2>]
```

The <port | port1-port2> option lets you specify a single port or a range of ports.

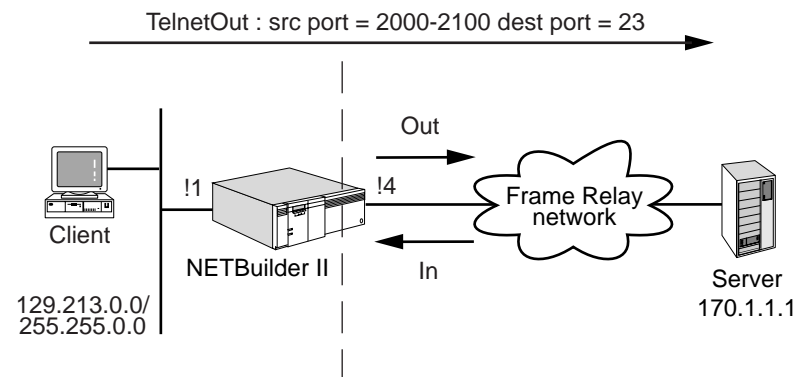
For example, you can define a service by entering:

```
ADD -FireWall UserDefService my_telnet TCP Src > 1024 Dst = 23
```

The service name can then be used in place of the <protocol> option when defining rules. After a service is defined, the Src and the Dst ports value can be interchanged by using !serviceName. For example, using !my\_telnet, then means TCP Src = 23 Dst > 1024.

**Example** The example in Figure 130 shows how to configure a service and then how to use the service name when you define packet filtering rules. Assume that there is no predefined filter for Telnet and a client wants to telnet to the server.

**Figure 130** Defining a Service



To configure the Telnet service and define packet filtering rules, add a service name by entering:

```
ADD -FireWall uds my_telnet TCP Src 2000-2100 Dst = 23
```

If you specify my\_telnet as the name when defining packet filtering rules, it means that the user protocol is TCP, destination port is 23, source port can be in the range 2000-2100, the source is allowed to initiate a connection.

### Configuring a Firewall Using a User Defined Service Name

You can then use service rules to create a firewall. Firewall configuration should be done on port 4 to allow a Telnet session between the client and the server (initiated from the client) using:

```
permit my_telnet from 129.213.0.0/16 to 170.1.1.1 (on the output filter)
permit !my_telnet from 170.1.1.1/16 to 129.213.0.0/16 (on the input filter)
```

### Differences Between a Predefined Service Filter and a User Defined Service Based Filter

There is a difference between the user defined service filter and predefined service filter in terms of the number of rules to be applied on a port. For example, if TelnetOut was allowed in Figure 27, the following predefined service filter would have to be used:

```
ADD !4 -FireWall TELnetOut Permit From 129.213.0.0/16 to 170.1.1.1
```

This filter would have to be used as the output filter rule and it would have internally created another rule in the input direction to permit the traffic from 170.1.1.1 to 129.213.0.0/16. Using the predefined filter eliminates the need to create an additional rule.

With the user defined service, you would need to create two rules in the input and output directions. However, the user defined service is useful when creating filters for protocols for which no predefined filters exist.

### Using IP Addresses Grouping

The ability to group addresses allows you to identify a group of IP addresses by name which need the same type of access. This name can then be used when defining filters.

Add the IP addresses to a list using:

```
ADD -FireWall AddressList <AddressListName> <IP Address/mask>
```

For example, to add an IP address to the address list named `my_list`, enter:

```
ADD -FireWall AddressList my_list 10.1.1.10
```

You can then create rules by entering:

```
permit from my_list to 10.1.1.1
```

and

```
permit from 10.1.1.1 to my_list
```

Assuming that `my_list` has the three IP addresses 20.1.1.1, 30.1.1.1, 40.1.1.1, the following three rules are created internally:

```
permit from 10.1.1.1 to 20.1.1.1
permit from 10.1.1.1 to 30.1.1.1
permit from 10.1.1.1 to 40.1.1.1
```

The last rule corresponds to "permit from 10.1.1.1 to `my_list`"

If you define a rule by entering:

```
permit from my_list to my_list
```

The following rules are created internally:

```
permit from 20.1.1.1 to 30.1.1.1
permit from 20.1.1.1 to 40.1.1.1
permit from 30.1.1.1 to 20.1.1.1
permit from 30.1.1.1 to 40.1.1.1
permit from 40.1.1.1 to 20.1.1.1
permit from 40.1.1.1 to 30.1.1.1
```

### Using the RealPlayer Predefined Service

The RealPlayer application includes the RealAudio and RealVideo applications. This section describes the operation of RealAudio; operation of RealVideo is same as the RealAudio.

To support RealAudio and RealVideo through the firewall, the traffic should be enabled on the following range of ports:

- TCP port 7070 for the incoming and outgoing traffic
- Negotiated UDP port during TCP session setup, for the incoming traffic from the server

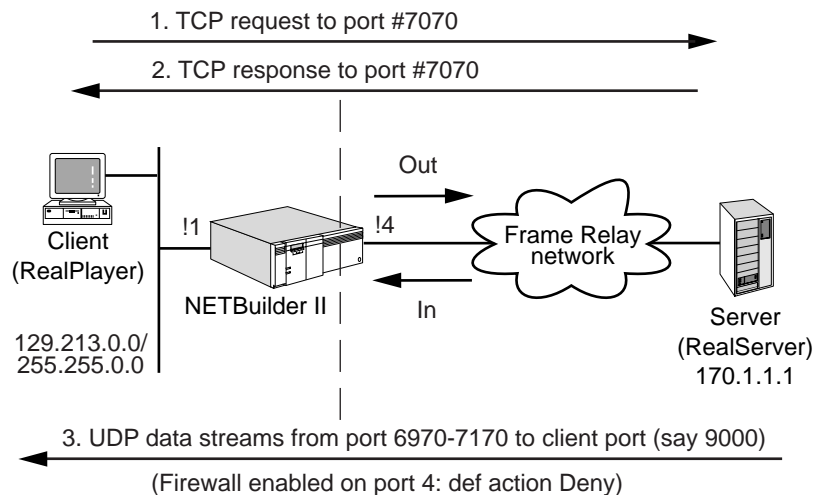
The TCP port is used by the client (RealPlayer) to initiate a conversation with an external RealServer, to authenticate the player to the server, and to pass control messages during playback including pausing or stopping the audio stream.

The range of UDP ports, on the other hand, carry the incoming stream. These ports begin to carry traffic only after the client and server have performed the authentication routine, and should be enabled only for incoming traffic. The client's UDP port to which this traffic should be sent is made known to the server during TCP packet exchange.

The RealAudio protocol sequence between the client and server is shown in Figure 131. The protocol sequence steps are as follows:

- 1 The client (RealPlayer) sends initial TCP request to the server (RealServer) with source port 7070 and destination port 7070.
- 2 The RealServer sends TCP response back to the RealPlayer with source port 7070 and destination port 7070.
- 3 The connection is established.
- 4 If the RealPlayer decides to get the audio streams delivered at its UDP port, the RealServer sends the UDP stream from any of the ports greater than 1023 to one of the client's ports, which the client would have identified during the initial TCP packets exchange.
- 5 If the RealPlayer decides to get the audio streams delivered at its TCP port, the RealServer sends the TCP streams from its port 7070 to the client port 7070.

**Figure 131** RealAudio Protocol Sequence



### Firewall Configuration

Using Figure 131 as an example, you can configure the FireWall Service to permit users inside a private network to access the RealServer. This configuration does not allow RealServer to initiate a connection to the RealPlayer. Since the FireWall Service is not configured on port 1, there are no restrictions on the traffic that is allowed to pass through the port. However, since the FireWall Service is configured on port 4 and the default action is deny, the following output filter rule should be added on port 4:



```
Add !4 -fw RealPlayer permit from 129.213.0.0/16 to 170.1.1.1/16
ClientUDPPort 9000
```

When this rule is added, it will internally create the following filters to support the RealAudio data flow:

- An output filter to permit TCP connection from client to server:  
`Permit from 129.213.0.0/16 to 170.1.1.1 TCP Src=7070 Dst =7070`
- An input filter to permit TCP connection from RealServer to RealPlayer  
`Permit from 170.1.1.1 To 129.213.0.0/16 TCP Src=7070 Dst =7070 Estab`
- An input filter to permit UDP streams from server to client  
`Permit from 170.1.1.1 To 129.213.0.0/16 UDP Dst = 9000`

### Additional Predefined Services

The following sections describe support for predefined services.

**TraceRoute Support** To allow the traceroute outbound where the user on the Internal network is running traceroute to the external destination through the FireWall Service, the following packets are allowed through the firewall:

- Constructed UDP packets outbound.
- Relevant ICMP packets ("TTL exceeded" and "port unreachable") back inbound.

The NETBuilder bridge/router generates a TraceRoute packet by creating a UDP packet and using 33434, 33435, 33436 as the destination port numbers. The source port is created randomly.

**DNS Client to Server Support** To allow the DNS query outbound where user on the Internal network sends DNS query to the external destination through the FireWall, the TCP/UDP packets at the following ports are allowed through the FireWall:

- TCP/UDP packets from the client to the server with Dst 53
- TCP/UDP packets from the server to the client with Src = 53 Dst = client's port specified in the request

**Secure HTTP (S-HTTP/HTTPS) support** The HTTPS protocol is HTTP on top of Secure Socket Layer (SSL). SSL was designed by Netscape and specifies a protocol for providing data security layered between application protocols (such as HTTP or NNTP) and TCP/IP. This security protocol provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection.

SSL was designed to provide security between client and server and to avoid any kind of 3-way man-in-the-middle attack. Therefore SSL cannot be proxied through traditional application level firewalls, because SSL considers a proxy server to be a middleman. The simplest alternative to this problem is to use a packet filtering firewall. The SHTTP server listens at the TCP port 443, hence in order to allow the SHTTP traffic to pass through the firewall, the traffic destined to and from port 443 should be allowed unrestricted through it.

**BGP-4 Support** BGP is a TCP based protocol with no client-server relationship (it has a peer-to-peer relationship). Each of the BGP peers listens on the TCP port 179. When a BGP peer wants to establish a TCP connection, it uses port 179 as the destination port; the source port can be anything.

**Finger Support** The Finger Service provides information about users like person's real name and a user name login time. The user may want to limit incoming finger requests. Outgoing finger requests are mildly problematic. The attacks are possible through the data in the finger response. The attack can be in the form of server sending immense data to the client or sending certain control characters to reprogram client's keys or sending a command that mails the password file to the server. You should be aware of these consequences which may happen due to allowing finger in or out.

**whois Support** *whois* is commonly used to obtain public information about hosts, networks, and domains. Sites generally do not provide their own whois server; they merely access the whois servers at the NICs. There have been no known security problems with whois clients. Security problems that have occurred are data driven.

**SOCKS Support** SOCKS is a networking proxy protocol that enables hosts on one side of the SOCKS server to gain full access to hosts on the other side of the SOCKS server without requiring direct IP reachability. SOCKS redirects connection requests from hosts on opposite sides of a SOCKS server. The SOCKS server authenticates and authorizes the requests, establishes a proxy connection, and relays data. The SOCKS server listens on the standard TCP port 1080, however SOCKS can be configured to listen to some other port as well.

**Enabling Security** To allow a predefined service access through the firewall, use:

```
Add !<port> -FireWall <pre-defined service name> permit | deny [Log
[0-7]] [From <IPaddr/mask>] [To <IPaddr/mask>] [NextHop <IPaddr>] [Secure
| NoSecure | Both]
```

To allow secured telnet traffic to come in (only) through the firewall, enter:

```
Add !1 -FireWall IPSecESP permit from 10.1.1.1 to 20.2.2.2
```

To allow only packets with ESP headers through the firewall, enter:

```
Add !1 -FireWall TELnetIn permit from 10.1.1.1 to 20.2.2.2 Secure
```

This command allows only secure TelnetIn packets in through the firewall.

To allow both Telnet in and Telnet out packets through the firewall, you must also enter:

```
Add !1 -FireWall TELnetOut permit from 10.1.1.1 to 20.2.2.2 Secure
```

Each predefined service parameter must be specified individually.

---

## Managing Filters

This section describes the syntax for creating filters, how to create and delete filters, how to manage filters in your firewall configuration, and the differences between traditional IP filters and firewall filters.

**Filter Rule Syntax** For detailed information on rule syntax and corresponding values, see “Filters” in the FireWall Service Parameters chapter in *Reference for Enterprise OS Software*.

**Creating Filters Using Filter Rules** Each filter has a name assigned to it. The syntax of a filter name is the same as a DOS filename; it can be up to eight characters followed by up to a three-character extension. File names are case-insensitive. When a new filter is created with the same name as an existing filter, the new filter replaces the old one, in memory and on the disk.

A filter must have at least one rule defined within it. Each rule must begin with one of two keywords: Permit or Deny. There is no limit to how many rules can be defined in a filter; the software continues to accept new rules as long as there is memory or disk space available for them.

**Defining a Filter Using the ADD Filter Command**

To define a filter use:

```
ADD Filter <filter name> [<rule 1> <rule 2> <more rules> ...]
```

A filter begins with the left parenthesis, and terminates with the right parenthesis. In between the parenthesis, any number of rules are allowed. Empty rules are ignored.

Each rule must be terminated by a new-line character or multiple rules may be specified on one line when each rule is terminated by a semicolon (;). Rule syntax checking is performed when rules are entered. Any syntax error is identified immediately and triggers the following actions:

- Displays error descriptions about the kind of syntax problem
- Discards the current line of input
- Displays help information
- Prompts for continued input

A rule that begins with the pound (#) sign indicates that it is a comment, and the software ignores the whole line.

**Creating Filters Using An Off-line Editor**

You can use any PC or workstation text editor to create and edit your filter files. The workstation is also a good place to back-up the filters file. Because filters can be complex, do the following:

- Examine and carefully edit the filters on your workstation
- Add comments to them using the pound (#) sign
- Use TFTP to transfer the final results under the FILTERS directory on the bridge/router.

After the file transfers are complete, you must either reboot the bridge/router or issue the REStart command to restart the firewall with the newer set of filters.

Use the TESt command to test filters with test packets generated by the bridge/router. The system then reports whether the packet was permitted or denied.

The REStart command examines the filter file, detects any syntax errors, and provides the line number, the offending keywords, and other applicable help information. If there is a syntax error in the filter file, none of the defined filters

will take effect. As a result, you are responsible for making appropriate corrections to the filter file off-line and reentering the REStart command.

A filter file must contain only filter rules conforming to the syntax specified in “Filters” in the FireWall Service Parameters chapter in *Reference for Enterprise OS Software*. Blank lines and comment lines starting with the pound sign (#) are ignored.

**Displaying Filters** To display all the filters that are currently defined, enter:

```
SHow -Firewall Filter
```

The display shows the names, sizes, and creation dates for all of the filters that are currently stored on disk.

To display the contents of a particular filter, use:

```
SHow -Firewall Filter <filter name>
```

**Deleting Filters** Filters must be individually deleted from the system.

To delete a filter, use:

```
DElete -Firewall Filter <filter name>
```

Deleting a filter only removes it from local storage but not from memory. To remove it from memory, run the restart command.

**Assigning Filters to Interfaces**

Each interface (port) can have two filters associated with it: one filter that applies to traffic received on that port, and one that applies to traffic to be transmitted on that port.

The command syntax for InFilter and OutFilter is:

```
SETDefault !<port> -Firewall InFilter = <filter name>
```

```
SETDefault !<port> -Firewall OutFilter = <filter name>
```

For more information on the InFilter and OutFilter parameters, see those sections in the FireWall Service Parameters chapter in *Reference for Enterprise OS Software*.

All incoming packets, including broadcast, unicast, and multicast IP packets, received on the <port> are subject to filtering operations as specified in the InFilter parameter. Packets going to the router, as well as packets to be forwarded by the router, are all subject to filtering, including all the incoming routing protocol packets such as OSPF, RIP, or others.

All outgoing packets, including broadcast, unicast, and multicast IP packets, to be transmitted over <port> are subject to the filtering operation as specified in the OutFilter parameter. Packets originating from the router (for example, routing protocol packets), as well as forwarded packets (from another source), are all subject to the same filtering operations.

To remove a filter from the interface, use:

```
SETDefault !<port> -Firewall InFilter = "" (empty name)
```

If an assigned <filter name> is not found in local storage, no filter operation is activated.

### Activating and Deactivating Filters

To enable or disable all firewall filtering on an interface, use:

```
SETDefault !<port> -FireWall CONTrol = Filter | NoFilter
```

Each interface can be enabled independently of other interfaces. Enabling filtering on an interface enables ALL of the applicable firewall filters including:

- Service-independent filters, such as SourceSpoofing and IPTunnel.
- Generic filters, such as those defined in the InFilter and OutFilter parameters.
- Default actions as defined in the DefAction parameter and predefined filters.

Only packets being forwarded by IP forwarding routine, as well as those destined to or originated from local system, are subject to the filtering action; packets being bridged are not subject to this service. Bridging filters are the appropriate place to define rules for filtering bridged packets.

### Firewall Filters versus IP Filters

There are several differences between firewall filters and traditional IP filters used on the NETBuilder II and SuperStack II bridge/routers:

- Traditional IP filters are output filters only. They cannot be configured to perform input filtering, an important first level of defense. Not only is input filtering necessary in blocking certain attacks (such as IP source address spoofing), it also protects the router itself.
- Separate filters cannot be specified on a per-interface basis using traditional IP filters.
- Traditional IP filters contain non user-friendly hex numbers, offsets, and masks. This syntax is complex and confusing for users to design and maintain.
- Traditional IP filters cannot effectively deal with packets containing IP options.
- Traditional IP filters reorder the filtering rules to their own preference. Users may be confused in some situations, and the filter design may be defeated in other situations.

Because of the special role traditional IP filters play, they remain unchanged and operate in parallel with the firewall. Some specific services are only available from the traditional IP filter such as X.25 profile ID, packet priority, protocol reservation, and dial-on-demand discard. For those services, users must continue to use existing IP filters.

### Filters — Firewall Execution Order

When IP and Firewall filters are enabled, the sequence of execution for through traffic is as follows:

- receive packet -> input firewall filter -> "forwarding decision" -> traditional IP filter -> output firewall filter -> transmit packet

For traffic coming into and terminating at the bridge/router, the sequence is:

- receive packet -> input firewall filter -> "internal process" (no IP filter is applied)

For traffic originating from within the bridge/router, the sequence is:

- "internal process" -> output firewall filter -> transmit packet

Traditional IP filters only apply to “through” traffic. Traffic *to* the router is unfilterable.

## Setting Up System Logs

IP Firewall can be configured to log system messages to the AuditLog Service, to the local console, or both. Log messages contain crucial information such as the date and time, interface, incoming and outgoing, packet header summary, and reason. Each message contains two types of codes: facility and priority. The facility code tells syslog what subsystem the message is from and the priority code tells syslog how important the message is (ranging from Log(0), emergency, which is the highest priority to Log(7), debug, which is the lowest priority). Logging can be done on permitted or denied packets.

You can set up your firewall to log system messages in one of three ways:

- If you want log messages sent to your local console port, enter:

```
SETDefault -FireWall Log = Console
```

- If you want log messages sent to your the AuditLog service, enter:

```
SETDefault -FireWall Log = Syslog
```

The AuditLog Service controls the delivery of messages to the syslog server(s). Most UNIX workstations come with syslog support. Consult your workstation manual for more details. If you choose this logging method, you need to enable the AuditLog Service and configure the IP address of your syslog server, using:

```
SETDefault -AuditLog CONTROL = (CONfig, MESSAGES, SECURITY)
SETDefault -AuditLog LogServerAddr = <IP address>
```

- If you want to send log messages to your local console port and to your syslog server, enter:

```
SETDefault -FireWall Log = (Syslog, Console)
```

## Specifying Log Content

You can enable specific information to be logged. For example, to see all of the denied source-spoofing packets, enter:

```
SETDefault -FireWall Log = SrcSpoofing
```

If you want to log all incoming FTP connections, add a Log option to your -FireWall FTPIn command. For example:

```
ADD !2 -FireWall FTPIn Permit Log
```

You can log the summary of the packet or detailed contents (the first 64 bytes) of the packet using:

```
SETDefault -FireWall Log = Summary | Detail
```

For more information on selectively logging messages, see the Log parameter in the FireWall Service Parameters chapter in *Reference for Enterprise OS Software*.

## Log Description

The log generated by the firewall has the following format:

```
<Date, time> Firewall <Tx | Rx> | !<interface> <Source IP addr>
<Source Port> -> <Destination IP addr> <Destination port> <Protocol>
<action> <Filter type>
```

The fields in the log contain the following information:

<Date, time>	Specifies the actual date and time when the firewall received or transmitted a packet
<Tx   Rx>	Specifies whether the packet was being transmitted or received through the firewall. Tx indicates transmitted, Rx indicates received.
!<interface>	Indicates the interface number on the NETBuilder bridge/router on which the packet filtering rules were applied for the packet. In the receiving direction, the interface number is the interface on which the packet was received. In the transmitting direction, it represents the interface on which the packet was transmitted.
<Source IP addr>	Indicates the source IP address of the packet.
<Source port>	<p>Indicates the source port for the TCP/UDP packet. For the well known applications, the name of the application is specified instead of actual TCP/UDP port numbers. For example, if there is an incoming telnet packet received by the firewall, the log will say "Telnet" instead of 23 (which is the actual TCP port for the telnet application).</p> <p>However, with the increasing use of random UDP ports by applications (multimedia applications, Archie services, DNS services etc.) it may not be possible for the firewall to find out the type of traffic running through the port.</p> <p>For example, if a user is running RealAudio traffic through the firewall which uses destination TCP port 7070 (for the server) and any random TCP source port above 1023, and if that port happens to be 1525 (which is the well known port for Archie), the log will indicate that the Src port is Archie and Dst port as RealAudio. When such confusion occurs, the user should analyze the previous log history to find out the kind of traffic going through their firewall.</p>
<Destination IP addr>	Indicates the destination IP address for the packet.

<Destination port>	<p>Indicates the destination port for the TCP/UDP packet. For the well know applications, the name of the application is specified instead of the TCP/UDP port number. For example, if there is an outgoing telnet packet transmitted through the firewall the log will say "Telnet" instead of 23 which is the actual TCP port for telnet. However, with the increasing use of random UDP ports b the applications (multimedia applications, Archie services, DNS services), it may not be possible for the firewall to find out the type of traffic running through it.</p> <p>For example, if a user is running Real Audio traffic through the firewall which uses TCP port 7070 and any random UDP port above 1023, and if the user specifies pot 1525 as the UDP port on which to receive the RealAudio stream, the log will indicate the destination port as Archie and the Src port as RealAudio. When such confusion occurs, the user should analyze the previous log history to find out the king of traffic going through the firewall.</p>
<Protocol>	<p>Indicates the protocol carried within the IP packet. TCP/UDP/ICMP/GRE. AH/ESP protocols are specified by name. Any other protocols are specified by numeric value. For ICMP packets, this field is followed by the ICMP packet type.</p>
<action>	<p>Indicates whether the firewall permitted or denied the packet. Permit means the packet was allowed to pass through the firewall, Deny means that the packet was dropped by the firewall.</p>
<Filter type>	<p>Indicates the type of firewall filter which was applied to the packet when making the permit/deny decision. For example, if you configured a TelnetIn service filter and the incoming IP packet matches with this rule, then the log will indicate TelnetIn. Other keywords include DefActionIn, DefActionOut, InFilter, and OutFilter. However, for the DNSClientSvrIn, DNSClientSvrOut, RealPlayer, TraceRouteIn, and TraceRouteOut services, this filter type will always be either InFilter or OutFilter.</p>

### Firewall Log Examples

The following example shows that the firewall received a packet on port 4 from 10.0.0.2 which was addressed to 10.0.0.255 and it was a RIP packet. The protocol was UDP. The firewall permitted this packet and this action was taken as a result of DefActionIn.

```
[161] NETBuilder # May 4 10:16 FireWall Rx !4 10.0.0.2 (RIP)->10.0.0.255
(RIP) UDP Permit DefActionIn
```



The following example shows that the firewall received a packet on virtual port 1 with the source address of 129.213.205.110 and with a DNS source port 53. The destination address was 129.213.119.38 and the destination port was 32877. The packet was permitted through the firewall as a result of InFilter rules.

```
May 5 13:54 FireWall Rx !V1 129.213.205.110 (DNS)-> 129.213.119.38 (32877)
UDP Permit InFilter
```

Additional log examples include the following:

```
May 5 13:17 FireWall Tx !6 10.0.0.2(20522)->10.0.0.1(1723) TCP Permit
OutFilter
May 5 13:58 FireWall Tx !V1 129.213.119.10 (RIP)->255.255.255.255(RIP) UDP
Permit RIP
```

## How a Firewall Works

The firewall allows users inside a private network to have outbound access, while restricting outside users from inbound access. The types of incoming and outgoing traffic can be identified as follows:

- Inside-originated request to an outside service
- Outside reply to the inside-originated request
- Outside-originated request to an inside response
- Inside reply to the outside request

Firewalls are typically constructed on bastion hosts and a multiprotocol router. The bastion hosts, usually UNIX, are configured to prevent it from being compromised by outsiders and to provide detailed logging of system activity for security monitoring. The host may serve as an externally accessible server for FTP, e-mail, or the WWW.

Another common firewall component is a packet-filtering router. The multiprotocol router has extensive filtering capabilities to limit the type and direction of traffic that passes through it. The router usually is not the object of an attack, but can serve as a barrier to other, more desirable targets, or as the basis of a denial-of-service attack.

## Packet-Filtering Routers

The packet-filtering router can make a permit or deny decision for each packet it receives. The router examines each datagram to determine if it matches one of its packet filtering rules. The filtering rules are based on the packet header information that is made available to the IP forwarding process. This information consists of the following items:

- IP source address
- IP destination address
- Incoming interface of the packet
- Outgoing interface of the packet
- Encapsulated protocol (TCP, UDP, ICMP, or IP tunnel)
- TCP/UDP source port
- TCP connection Establishment packets
- TCP/UDP destination port

- ICMP message type

If a match is found and the rule permits the packet, then the packet is forwarded according to the information in the routing table. If a match is found and the rule denies the packet, then the packet is discarded, unless it is redirected by either NextPort or NextHop. If there is no matching rule, a user-configurable “default action” parameter determines whether the packet is forwarded or discarded.

### Benefits of Packet-Filtering Routers

The majority of Internet firewall systems use only a packet-filtering router. Other than the time spent designing the filters and configuring the router, little or no cost is required to implement packet filtering because the feature is included as part of standard router software releases. Because Internet access is usually provided over a WAN interface, there is little impact on router performance if traffic loads are moderate and few filters are defined. A properly designed firewall using a packet-filtering router can be transparent to end users and applications, so it does not require specialized user training or require that specific software be installed on each host.

---

## Firewall Filter Types

You can configure your firewall with the following three types of filters:

- Service-independent filters
- Predefined (service-dependent) filters
- Generic filters

This section describes each type of filter and how each can be used as a component of a firewall.

### Service-Independent Filters

Some types of Internet attacks are popular, but difficult or impossible to specify using only generic packet-header information. These attacks are generally service-independent and are difficult to specify because filtering rules require additional information that can only be learned by looking in the routing table. For these types of attacks, a separate control (-FireWall CONTROL) has been created for each of the known attack types. The -FireWall CONTROL parameter works at the IP layer and allows you to control the following items:

- Filtering
- Source IP address spoofing
- TCP/IP tiny fragment attacks
- Packets that contain IP options such as source-route, record-route, and time-stamp
- IP-over-IP tunnels
- ICMP messages (protection against denial of service attacks)

For more information on the -FireWall CONTROL parameter, see the Firewall Service Parameters chapter in *Reference for Enterprise OS Software*.

### Predefined (Service-Dependent) Filters

This section identifies some of the important services commonly used over Internet (Telnet, SMTP, NNTP, FTP, HTTP, Gopher, and DNS) and describes how the parameters in the Firewall Service relate to each service. These parameters are

designed for filtering that service; one parameter controls incoming connections and one parameter controls outgoing connections. Each parameter can be separately configured on a per-interface (per-port) basis.



*The filtering operations performed by these service-dependent filters can also be performed using generic filters, which are described in “Generic Filters” later in this chapter.*

Service-dependent filters, and their related parameters, provide the following benefits:

- The parameters are designed for connection flows instead of packet flows. Connection flows deal with issues such as outbound FTP sessions or inbound Telnet sessions. Packet flows (the basic concept of generic filter rules) deal with inbound TCP packets or outbound UDP packets. A connection flow (such as inbound Telnet) usually involves bidirectional packet flows (such as both inbound TCP packets and outbound TCP packets).
- These parameters, such as FTPOut, automatically take care of both outbound and inbound TCP packets.
- These parameters allow a user who is not an expert on packet formats, or who may not be aware of the protocol details and port number schemes, to easily configure a packet-filtering router.
- Some services, such as DNS, are quite complex and they are difficult to specify using the generic filter rules.

Be aware that providing packet-filtering rules for a particular service does not mean the service is secured. Each service has its own weaknesses and security holes that are beyond the ability of a packet-filtering router to control. In general, permitting outbound connections is very safe. Outbound connections are connections initiated from internal networks to the Internet; they do not permit Internet-initiated connections into internal networks. Permitting inbound connections are much more risky. Consider using application-level proxy services that will further enhance your firewall.

TCP-based services such as TELnet, FTP, SMTP, NNTP, DNS, Gopher, and Archie are much safer than non-TCP-based services. Avoid using non TCP-based services on your Internet connections. Because of the CPU requirements of DenySrcSpoofing, it is recommended that it be turned on only where absolutely required. Interfaces receiving external traffic should have DenySrcSpoofing turned on to repel source spoofing attacks. Internal interfaces may not require this type of checking.

### **Dynamic “Window Management” for FTP**

FTP is a TCP-based service, and is unusual because it uses two or more simultaneous TCP connections. The first TCP connection is initiated from client to server. This connection, usually called the *command channel*, carries commands and replies. The second TCP connection, usually called the *data channel*, is dedicated to transferring data. The second TCP connection is made in two ways: regular FTP and passive FTP.

Regular FTP occurs when a client issues a PORT command on the command channel to the server and the server opens the data TCP session. The port number of the client’s desired data TCP socket is embedded in the PORT command.

Passive FTP occurs when a client issues a PASV command and, if the server responds positively, the client initiates the data TCP session. The TCP port number is embedded in the command and reply.

The FTPIn parameter and the FTPOut parameter understand both forms of FTP sessions. The FTP filters permit server-to-client data TCP connections when they detect the PORT command. They also permit a client-to-server data TCP connection when they detect a PASV command. The TCP port number is extracted from the PORT (or PASV) command so that only a specific data connection is allowed; no persistent holes in the firewall will occur.

An FTP session may involve several data TCP connections, therefore, the FTPIn parameter and FTPOut parameter constantly monitor the active command channels for PORT and PASV commands and readjust their permissions window accordingly.

When the command channel is closed, all associated data channels are also closed.

### Generic Filters

3Com's simple yet powerful filter language allows you to write your own specialized filters, each of which may be comprised of a number of rules within the generic filters. Some examples are:

- Rules can be specified on a per-interface basis.
- Rules can be applied to incoming traffic, outgoing traffic, or both.
- Rules are based on easy-to-understand names and values, instead of hex numbers, offsets, or bit-masks.
- Rules can provide comprehensive logging for both permitted or denied packets.
- Rules can permit or deny packets based on any combination of source address, destination address, protocols, source TCP/UDP port, destination TCP/UDP port, ICMP message types, and TCP "Establish" keyword to differentiate the direction of TCP connections from the value of the SYN bit.

---

### Firewall Terms

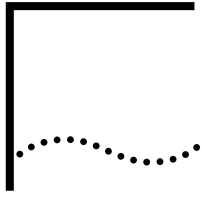
The following terms are used in this chapter to describe the firewall feature:

firewall	A router and/or workstation with multiple network interfaces that controls and limits specific protocols, types of traffic within each protocol, types of services, and direction of the flow of information.
secure logging	A method that takes an audit trail of system activity received from a bastion host and places it in a secure location.

## IP source address spoofing

Spoofing uses “forged” source addresses to make an outside packet appear to have come from the inside network so that the firewall allows it to have access to the private network. Spoofing works on the principle that, by default, routers perform route lookups only on the destination IP address in each packet, paying no attention to the incoming interface of the source IP address.





# CONFIGURING THE ACCESS CONTROL SERVICE USING RADIUS

This chapter describes how to configure the Access Control (AC) Service for Radius Access Control.

---

## Configuring AC

Telnet, Console or Weblink access authentication to the NETBuilder bridge/router can be accomplished locally by the NETBuilder bridge/router itself or by using a Radius Server.

The NETBuilder bridge/router performs local authentication by searching a local database for users and their passwords.

Radius authentication provides another means of authentication, utilizing an external Radius server with its own database of users, passwords, and access levels. This centralizes and simplifies the task of administering access to large number of routers on a network.

With Radius configured, when a user attempts to access the NETBuilder bridge/router, the user account information is sent to a Radius server. Based on the server response, the NETBuilder bridge/router takes the following actions:

- Declines access if the Radius account is invalid.
- Provides user or network manager level access depending on the users access level.
- If no response is received, tries a secondary Radius Server, if configured.
- If no response is received from the Radius Server(s) and a resolution order of "Radius Local" has been configured, the user is authenticated locally.

To set up Radius Authentication, follow these steps:

- 1 Specify the IP address of Primary Authentication server. For example, enter:

```
SETD -ACS PrimAuthSrvr = 192.147.72.84
```

- 2 Specify Resolution Order. For example, enter:

```
SETD -ACS RESolutionOrder = Radius
```

Resolution order sets the order for authentication, some variations on Resolution order follow:

If RESolutionOrder is set to "Radius Local", the NETBuilder bridge/router attempts to authenticate using the Radius server. If there is no response from the Radius server authentication is accomplished locally.

If RESolutionOrder is set to "Radius", the NETBuilder bridge/router authenticates using only the Radius server for users and network managers. Network manager

"root" overrides this and performs local authentication if there is no response from the Radius server. When setting up users on the Radius Server for network administrator privilege, select type "Administrative," for user privilege select service type "Login."

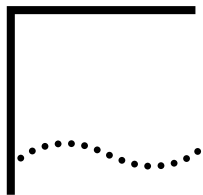


*All Radius parameters available in the AC menu are also available in the Remote Access Service (RAS) menu, when a common parameter is set in one service it is also set in the other service. The parameters are ACcntUdpport, AUthUdpport, PrimACcntSvr, PrimAUthSvr, RetransTimer, SecACcntSvr and SecAUthSvr.*



*At this time, Radius Accounting parameters can be set in the AC service but are only applicable in the RAS service. Only Radius Authentication is supported in the AC service and not Radius Accounting.*





# CONFIGURING THE LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL FOR FIREWALL CONFIGURATION

This chapter describes how to configure the Lightweight Directory Access Protocol (LDAP). The LDAP policy manager provides a transparent interface for components such as the Firewall component to interact with the LDAP client to store their policies on the LDAP server. This chapter describes how to configure your NETBuilder bridge/router to use Lightweight Directory Access Protocol (LDAP) to download the Firewall configuration from the LDAP server.

---

## Configuring LDAP

The Lightweight Directory Access Protocol defines a standard protocol for accessing diverse directory services. One use of a directory service is to store the security policies or the router configuration on the server and download these configurations when required.

The main advantage of centralizing the database is the reduction in the maintenance cost among others. Currently, the retrieval of user-based or the system-based firewall configuration from the LDAP server is supported on the NETBuilder bridge/router.

## Setting Up LDAP

This section describes the steps required to enable your NETBuilder bridge/router for LDAP. Before you begin the procedure, complete the following tasks:

- Log on to the system with Network Manager Privilege.
- Configure your ports and paths
- Configure IP routing

### Example

To enable your NETBuilder bridge/router for LDAP, follow these steps:

- 1 Configure the LDAP server address. For example, enter:

```
ADD -LDAP ServerAddress <Name | IP> [TCP port] <"service name">
```

You can assign the IP address or the domain name of the server. If the domain name is assigned, the name service configuration must be done using the IPN service parameters. The value of TCP port must be set to the value configured on the LDAP server where it listens to the LDAP requests. If the TCP port is not assigned, a default value of 389 will be used.

For example, to assign an IP address of 129.213.200.10 for the LDAP ServerAddress to fetch the Firewall configuration and use the default TCP port value of 389, enter:

```
ADD -LDAP ServerAddress 129.213.200.10 "firewall"
```

- 2 Configure the RootDN value. For example, enter:

```
ADD -LDAP RootDN <"DN string"> <"service name">
```

To assign a DN value of "o=3com, c=us", enter:

```
ADD -LDAP RootDN "o=3com, c=us" "firewall"
```

- 3 Configure the BindDN value. For example, enter:

```
ADD -LDAP BindDN <"DN string"> <"password"> <"service name">
```

To assign a DN value of "o=3com, c=us" and password as "3com", enter:

```
ADD -LDAP BindDN "o=3com, c=us" "3com" "firewall"
```

It is not necessary to follow the BindDN step. If this step is skipped, a NULL value is used.

- 4 Create UserProfile. For example, enter:

```
ADD -LDAP UserProfile <"user name"> <"service name"> <SysInit|LogOn>
<User|System>
```

To download the firewall policies for the user xxx at the time of logon, enter:

```
ADD -LDAP UserProfile "xxx" "firewall" LogOn User
```

To download the firewall system configuration for the user "system\_xxx", enter:

```
ADD -LDAP UserProfile "system_xxx" "firewall" SysInit System
```

- 5 Add the FireWall Service in the LDAP list. For example, enter:

```
Add -LDAP LocalOrLdap "firewall"
```

If the LDAP configuration is done for the first time on the NETBuilder bridge/router, the RESTART command must be run from the NETBuilder bridge/router command line. Running this command downloads all the firewall policies for the users configured in the UserProfile for which the value of SysInit/LogOn is set to SysInit.

## Verifying the LDAP Configuration

To verify the values associated with the LDAP, enter:

```
SHow -LDAP CONFIguration
```

## Adding Attributes and ObjectClasses on the Netscape Directory Server

For basic information, see "LDAP Terms" and "Extending Schema on LDAP Server" later in this chapter.

There is no standard way of loading the extended schema definitions on an LDAP server. Different server vendors may have different mechanisms to do this. These procedures assume the LDAP server to be a Netscape directory server. These procedures describe how to load the 3Com proprietary schemas on the Netscape directory server.

Steps to configure the Netscape directory server are out of scope for this section. However, procedures are provided to add the 3Com proprietary attributes and object classes as well as to populate the LDAP Server database.

### Adding Attributes

To add attributes, follow these steps:

- 1 Go to the Netscape Directory Server screen and click on the *Schema* button.
- 2 Type "policyname" in the Attribute Name window and click the Add New Attribute button to add the attribute "policyname."

- 3 Follow step 2 to add attributes "router-cmd" and "servicetype."

**Creating an Object Class** To create an object class, follow these steps:

- 1 Go to the Netscape Directory Server screen and click the *Schema* button.
- 2 Click Create Object class.
- 3 Type "policy" in the ObjectClass Name window.
- 4 Select "person" as the parent object class.
- 5 Add the attribute "policyname" and click the *Create New ObjectClass* button.  
This step creates a new object class called "policy"
- 6 Type "firewallpolicy" in the ObjectClass Name window.
- 7 Select "policy" as the parent object class.
- 8 Add the attributes "router-cmd" and "servicetype" and click the Create New ObjectClass button.

This step will create a new object class called "firewallpolicy."

### **Populating Database on LDAP Server**

Before downloading the firewall configuration from the LDAP server, the configuration must be added into the LDAP server database.

The first step towards populating the database is to select the directory information tree. The LDAP directory services organizes directory entries using a Directory Information Tree (DIT).

The design that the network/system administrator uses for the DIT is determined by the types of information that needs to be stored in the directory, for example, the physical nature of the enterprise and the applications that are used with the directory services.

As is the case with many directory services and file systems, the DIT is usually viewed as an inverted tree. The root of the tree is at the top and individual directory entries are at the lowest points on the tree. The root of the tree is represented by a special entry whose distinguished name (DN) is called the directory suffix. An important part of the DIT tree design is selecting this DN.

The best practice for choosing a directory suffix is to attempt to align it with a DNS name or internet domain name associated with an enterprise. This is the best approach regardless of the nature of the enterprise.

For example, when the system administrator wants to store the Firewall policy for each of its users in 3Com, the system administrator can define the base tree or the root DN.

The policies can be stored on the user basis. However, the tree can be defined in any form, for example, the policies can be stored on the basis of devices, locations, or IP addresses.

After the appropriate directory tree has been selected, the database can be added using the LDAP Data Interchange Format (LDIF) defined in the Netscape directory server.

## Creating Database in LDAP Data Interchange Format (LDIF)

LDIF consists of one or more entries separated by a blank line. Each LDIF entry consists of an optional entry ID, a required distinguished name, one or more object classes, and multiple attribute definitions. The basic form of an LDIF entry is:

```
[<id>]
dn: <distinguished name>
objectClass: <object class>
objectClass: <object class>
...
<attribute type>: <attribute value>
<attribute type>: <attribute value>
```

Only the DN and at least one object class definition are required. In addition, any attributes required by the object classes that are defined for an entry must also be defined on the entry. All other attributes and object classes are optional.

LDIF is a standard way of adding entries in the Netscape directory server. For more details, see the appropriate Netscape directory server manuals.

The following three entries describe the LDIF fields to add the firewall rule "add !1 -fw POPIn permit from 10.1.1.1" for the user called user10 in the tree whose root DN is set to "o=3com, c=us". The firewall rule will be applied to a fixed port, hence this rule should be downloaded at the system initialization time.

```
#Add user name user10
dn: cn=user10, o=3com, c=us
cn: user10
sn: user10
userpassword:user10
objectclass:top
objectclass:person
#Add a new policy called fwpolicy
dn: policyname=fwpolicy, cn=user10, o=3com, c=us
cn:user10
sn:user10
userpassword:user10
policyname: fwpolicy
objectclass:top
objectclass:policy
Add a firewall rule "add !1 -fw POPIn permit from 10.1.1.1"
dn: router-cmd=add !1 -fw POPIn permit from 10.1.1.1, policyname=fwpolicy,
cn=user10, o=3com, c=us
cn:user10
sn:user10
policyname:fwpolicy
router-cmd:add !1 -fw POPIn permit from 10.1.1.1
servicetype:firewall
objectclass:top
objectclass:firewallpolicy
```

The following four entries describe the LDIF fields to add the firewall rule "add !user\_xxx -fw rp permit from 30.1.1.1" and "add !user\_xxx -fw TraceRouteIn permit from 30.1.1.1" for the user called user\_xxx in the tree whose root DN is set to "o=3com, c=us". The firewall rule is applied to a dynamic virtual port. This rule should be downloaded at the user logon time.

```
#Add user name user_xxx
```

```

dn: cn=user_xxx, o=3com, c=us
cn: user_xxx
sn: user_xxx
userpassword:user_xxx
objectclass:top
objectclass:person
#Add a new policy called fwpolicy
dn: policymname=fwpolicy, cn=user_xxx, o=3com, c=us
cn:user_xxx
sn:user_xxx
userpassword:user_xxx
policymname: fwpolicy
objectclass:top
objectclass:policy
#Add a firewall rule 'add !user_xxx -fw rp permit from 30.1.1.1
dn:router-cmd=add !user_xxx -fw rp permit from 30.1.1.1,
policymname=fwpolicy,
cn=user_xxx, o=3com, c=us
cn:user_xxx
sn:user_xxx
policymname:fwpolicy
router-cmd:add !user_xxx -fw rp permit from 30.1.1.1
servicetype:firewall
objectclass:top
objectclass:firewallpolicy
#Add a firewall rule 'add !user_xxx -fw TraceRouteIn permit from
30.1.1.1'
dn:router-cmd=add !user_xxx -fw TraceRouteIn permit from 30.1.1.1,
policymname=fwpolicy, cn=user_xxx, o=3com, c=us
cn:user_xxx
sn:user_xxx
policymname:fwpolicy
router-cmd:add !user_xxx -fw TraceRouteIn permit from 30.1.1.1
servicetype:firewall
objectclass:top
objectclass:firewallpolicy

```

### General LDIF Rules

In general, to create a directory using LDIF on the Netscape directory server, follow these guidelines:

- Create an ASCII file containing the entries that need to be added in LDIF format. Each entry should be separated from the next by an empty line.
- Begin each directory in the database with the topmost, or the root entry. The root point of the directory must represent a suffix set for the directory server. The suffix actually specifies the distinguished name used for the local database. Incoming queries must have a suffix matching this value. For our example, this will be `o=3com, c=us`.
- When creating a directory, make sure that an entry representing a branch point is created before new entries can be created under that branch. In our example case, the following branch points should be created in order before storing the Firewall policies:

```

dn: cn=user10, o=3com, c=us
<list of attributes and object classes>
 dn: policymname=fwpolicy, cn=user10, o=3com, c=us
<list of attributes and object classes>

```

---

## How LDAP Works

On a typical LDAP model for an enterprise, the policies can be created offline on an NMS/workstation and stored on the LDAP server. The LDAP server can either be owned by an enterprise or it may reside outside an enterprise.

The network management station creates the security policies for its enterprise users and stores them on the directory server. The policies can later on be downloaded on the NETBuilder bridge/router controlling the various desktops. There are two possible ways in which these policies can be downloaded on the bridge/router:

- At the NETBuilder initialization time.

This approach is more suitable when the policies are group based (same policies for a group of people; not supported currently). In any case, the port/system based policies should be downloaded on the NETBuilder bridge/router at the time of initialization.

- When the remote user dials into the network.

This policy is more suitable when users have different security policies. When the remote user dials-in, the NETBuilder bridge/router does the authentication. The authentication information can either be stored locally or on the remote RADIUS server. When the user authentication is successful, LDAP server is contacted to get the firewall policies for the user. User authentication is not considered complete until firewall policies are obtained.

---

## LDAP Terms

*Attributes* Holds descriptive information about an entry. Each attribute consists of an attribute type and one or more attribute values. The attribute type identifies the class of information given by that attribute (e.g. telephone number). The attribute value is the particular instance appearing that entry (e.g. 326-7500). Attributes generally have short, mnemonic names, for example "cn" is the abbreviation for commonName.

*Object Class* Object classes define the type of data that an entry can contain (for example, does the entry represent a person, an organization, a country, and so forth). Each object class defines a set of required and optional attributes. This attribute list defines the kind of data that you can store on the entry.

*Schema* The type of information that can be stored in the directory is defined by the directory server's schema. The schema is defined by the following elements:

- Object classes
- Attributes

*Schema Extension* Schema extension involves defining new object classes and attributes to meet one's own needs to store specialized information. The schema supplied with the Standard directory server is limited and possibly cannot store the information you may want to store.

*Relative Distinguished Name (RDN)* RDN is the name of the entry itself, before the entry's ancestors have been appended to the string to form the full distinguished name.

*Distinguished Name (DN)* DNs are the string representation for the entry names in a directory server directory. The distinguished name of an entry is the string concatenation of the RDN's from the entry to the root of the tree.

### 3Com Proprietary Attributes and Object Classes Definitions

The schema of a directory defines the set of objects that can be created in that directory and the set of attributes that can be used to describe those objects. Extension of a directory schema is required when the standard schema supplied with the directory server does not meet a user's need. A schema can be defined by defining new attributes or by adding attributes to an object type or by defining a new object type.

To store the firewall configuration on the LDAP server, the following attributes and object classes have been defined. The newly defined attributes and the object classes are 3Com proprietary.

The attributes created are:

router-cmd, serviceType, policyName

The attribute definitions are as follows:

```
router-cmd ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX
 octetStringSyntax
:: = {attributeType 1}
servicetype ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX
 octetStringSyntax
:: = {attributeType 2}
policyname ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX
 octetStringSyntax
:: = {attributeType 3}
```

The object class created are:

policy, firewallpolicy

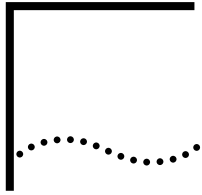
The definitions of the object classes are as follows:

```
policy OBJECT-CLASS
 SUBCLASS OF top
 MUST CONTAIN {
 policyName}
 MAY CONTAIN {
 description,
 }
:: = {objectClass 1}
firewallpolicy OBJECT-CLASS
 SUBCLASS OF policy
 MUST CONTAIN {
 router-cmd,
 serviceType}
:: = {objectClass 2}
```

The schema defined in this section should be used when accessing a directory via LDAP and searching or retrieving directory information for the FireWall Service.







# CONFIGURING REMOTE ACCESS SERVICES

This chapter describes how to configure the Remote Access Service (RAS) to enable remote users to connect to a central site in a virtual private network (VPN).

## Configuring Remote Access

This section describes the remote access configurations supported by NETBuilder bridge/routers.

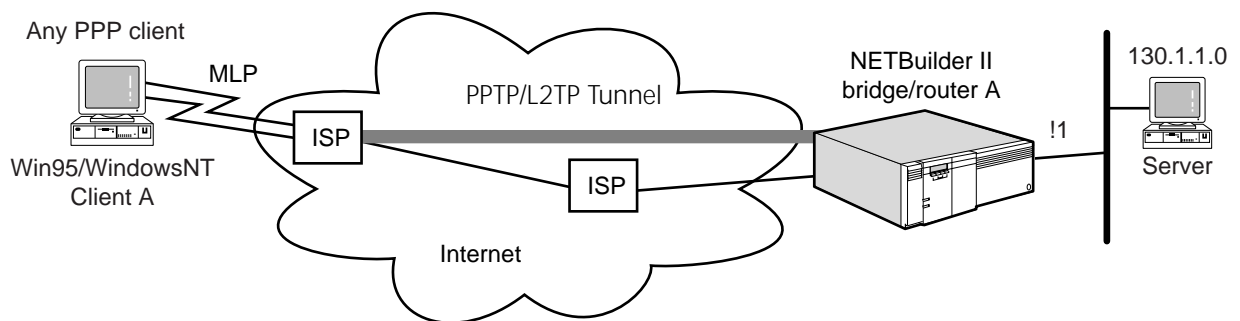
### Configuring Distributed Remote Access with a NETBuilder Tunnel Terminator

In a distributed remote access configuration, Windows 95/Windows NT client A dials into an Internet Service Provider (ISP) Point-of-Presence (POP) that can act like a Point-to-Point Protocol (PPTP) or Layer 2 Tunneling Protocol (L2TP) line server (LS). The LS at the ISP site sets up a PPTP/L2TP tunnel using the NETBuilder bridge/router as a tunnel terminator or packet processor (PP) and forwards all PPP packets from the Windows 95/Windows NT client A to the NETBuilder bridge/router using the PPTP/L2TP encapsulation. The PPTP/L2TP tunnel is transparent to the remote client.

A PPP session is set up directly between the remote client and the NETBuilder RAS server, and the client can run IP protocol to access the enterprise network. In this configuration, the client also makes a MLP connection with the ISP, and both PPP connections are bundled together at the NETBuilder RAS server.

The host network can be either an Ethernet network as shown in Figure 11 or a network running NetWare.

**Figure 132** Distributed Remote Access



*The configuration of the WAN port in NETBuilder II bridge/router A is not relevant to the RAS setup.*

### Prerequisites

Before beginning these procedures, complete the following tasks:

- Log on to the system with Network Manager privilege.

- Set up the ports and paths of your bridge/routers according to the instructions found in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- User Name and Password lengths are increased from 16 characters to 64 characters and the password length is increased from 8 characters to 32 characters.
- For multiple logins with the same User Name and Password, each login with the same user name is bound to one RAS port. The only exception is Multilink (MLP), when multiple logins can be bound to the same RAS port if:
  - The End Point Discriminators (EPDs) are the same.
  - MLP is selected and the EPDs are the same.



*For a connection to be considered an MLP, the MLP option must be selected and the EPDs must be specified.*



*When the host network is running NetWare, first follow the steps in the procedure "Configuring Distributed Remote Access for a NetWare Network," and then return to step 4 of this procedure.*

**Procedure** To configure distributed remote access, follow these steps:

- 1 Configure NETBuilder A LAN ports by entering:

```
SETDefault !1 -IP NETaddr = 130.1.1.1 255.255.255.0
```

The NETaddr parameter assigns an IP address to the specified port and configures the directly connected IP network or subnet.

- 2 Configure RAS IP.

- a Establish which IP network to use for RAS. RAS is supported on Ethernet LANs only. Therefore the IP network specified in this command must be an Ethernet network.

```
SETDefault -RAS IPNetwork = 130.1.1.0
```



*Specify only the IP network. A network mask is not required.*

- b Set up the IP pool option.

- When there is no IP pool enter:

```
SETDefault -RAS IPAddrPool = None (default)
```

- Or use the local DHCP server by entering:

```
SETDefault -RAS IPAddrPool = LocalDhcpServer
```

- Enter the range of IP addresses to be allocated by the DHCP server for RAS using:

```
ADD !<portlist> -DHCP AddressPool <IPaddr1> - <IPaddr2>
[!<profileid>]
```

- c Configure NetBIOS and DNS (for RAS) in DHCP using:

```
SETDefault !<profileid> -DHCP DNS
SETDefault !<profileid> -DHCP NetBios
```

- 3 Enable IP routing for RAS by entering:

```
SETDefault -IP CONTROL = Enable
```

#### 4 Configure the user database.

There are two ways to configure the NETBuilder bridge/router RAS user database. How the database is configured depends on where it will reside. The user database can reside either on the NETBuilder bridge/router (internal database) or on a RADIUS server (external database).

**d** To configure static virtual RAS ports for an internal database, follow these steps:

- Add virtual RAS ports using:

```
ADD !<vport> -Port VirtualPort RAS
```

- For each user, use:

```
ADD !<vport> -PPP AuthRemoteUser ("username", "password")
```

- Optionally, enable an authentication protocol using:

```
SETDefault !<vport> -PPP AuthProTocol = None | Pap | Chap | MS-Chap
```

- Enable the virtual port using:

```
SETDefault !<vport> -Port CONTrol = Enable
```

Each user requires a unique virtual port number, username and password.

Further, you may configure the idle timer value for each remote user, using:

```
SETDefault !<vport> - Port DialIdleTimer = <seconds>
```

**e** To configure the RADIUS server, follow these steps:

- Set the security type to RADIUS by entering:

```
SETDefault -RAS SecurityType = radius
```

- Set the primary authentication and accounting server to the designated RADIUS server using:

```
SETDefault -RAS PrimACcntSrvr = [<IP Address>]
```

```
SETDefault -RAS PrimaUthSrvr = [<IP Address>]
```

The RADIUS server is external to the NETBuilder bridge/router. When you set the RAS security type to radius, at a minimum you must configure the IP addresses for the primary authentication and accounting servers. For specific configuration details for your RADIUS server, consult the manufacturers documentation.

- Optionally, set the secondary authentication and accounting servers to the designated RADIUS server using:

```
SETDefault -RAS SecACcntSrvr = [<IP Address>]
```

```
SETDefault -RAS SecAUthSrvr = [<IP Address>]
```

- Set the secret string on the NETBuilder to match the RADIUS server's RADIUS Client. The default value is "3Com" for the NETBuilder II bridge/router.

- Make sure the UDP ports for authentication (default value 1645) and accounting (default value 1646) match the values defined in the RADIUS server. If these values do not match you can change the values in the NETBuilder bridge/router using:

```
SETDefault -RAS AuthUdpport = <UDP port number>
```

```
SETDefault - RAS ACcntUdpport = <UDP port number>
```

When using an external user database such as a RADIUS server, ports are configured dynamically as they are needed. You do not need to configure any ports.

- 5 Optionally, configure a port use limit, using:

```
ADD -PO PortLimit RAS <minimum> <maximum>
```

You can limit the number of ports that are used for RAS connections. By default there is no limit other than the limit to the number of possible virtual ports in your configuration. Using this command you can guarantee that a minimum number of ports will be available for RAS connections, and no more than your specified number of ports will be used for RAS connections.

- 6 Optionally, add RAS traps to the list of traps sent to the SNMP Network Manager, using:

```
ADD -SNMP TrapProfile "<TrapProfileName>" REMote
```

- 7 Enable RAS by entering:

```
SETDefault -RAS CONTROL = Enable
```

- 8 Configure the PPTP/L2TP tunnel by following these steps:

- a Enable the PPTP or L2TP function using:

```
SETDefault -L2T CONTROL = Enable Protocol = PPTP
```

or

```
SETDefault -L2T CONTROL = Enable Protocol = L2TP
```

- b Add an access list entry for the IP address of a PPTP/L2TP tunnel. Make sure flow control is enabled using:

```
Add -L2Tunnel AccessList <IPAddress> [<Network Mask>] [Protocol = PPTP]
[FlowControl=<Enabled | Disabled>]
```

Flow control applies to PPTP tunnels only. The IP address and the network mask are set to that of a PPTP/L2TP tunnel terminator. In this configuration, this IP address is the remote ISPs IP address. For more information about configuring PPTP/L2TP tunnels, see the Configuring L2Tunnel Connections chapter.

- 9 If the L2TP is used as the tunneling protocol, you need to set the local user name of the NETBuilder bridge/router using:

```
SetDefault -L2Tunnel L2TPLocalUser = ("<userid>", "<password>")
```

The assigned user ID is used as the "host name," which is required by the L2TP protocol during tunnel establishment.

### Configuring Distributed Remote Access for a NetWare Network

When the host network is running NetWare, enable distributed remote access by following these steps:

- 1 Configure NETBuilder A LAN ports by entering:

```
SETDefault !1 -IPX NETnumber = &12345678 Ethernet
```

The NETnumber parameter assigns an IPX network number where the NetWare server resides.

- 2 Configure RAS IPX.

Establish which IPX network to use for RAS.

```
SETDefault -RAS IPXNETWORK = &myNet00
```

- 3 Enable IPX routing for RAS by entering:

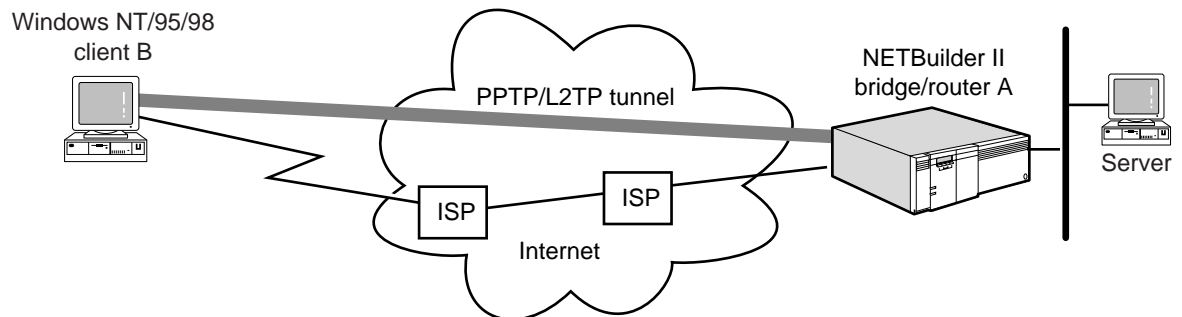
```
SETDefault -IPX CONTROL = Enable
```

You should now continue to configure the NETBuilder bridge/router by following the steps in the previous procedure starting at step 4, "Configure the user database."

### Windows 95 with Dialup Networking 1.2 or Windows 98 or Windows NT 4.0 Client Options

In the configuration shown in Figure 12, the Windows 95 client B has the Dialup Networking 1.2 upgrade, or Windows 98, or Windows NT 4.0. Client B first makes a dial-up connection to the ISP site to gain Internet access. Since the client is using an IP address assigned by the ISP site, it is generally denied access to the enterprise network. To gain the access to the enterprise network, a PPTP/L2TP tunnel is established between the client and the NETBuilder RAS server. In this configuration, the PPP connection is initiated by the remote client software. IP protocol can then be used over the PPTP/L2TP tunnel.

**Figure 133** Using Windows95 Dialup Networking 1.2, Windows98, or Windows NT 4.0

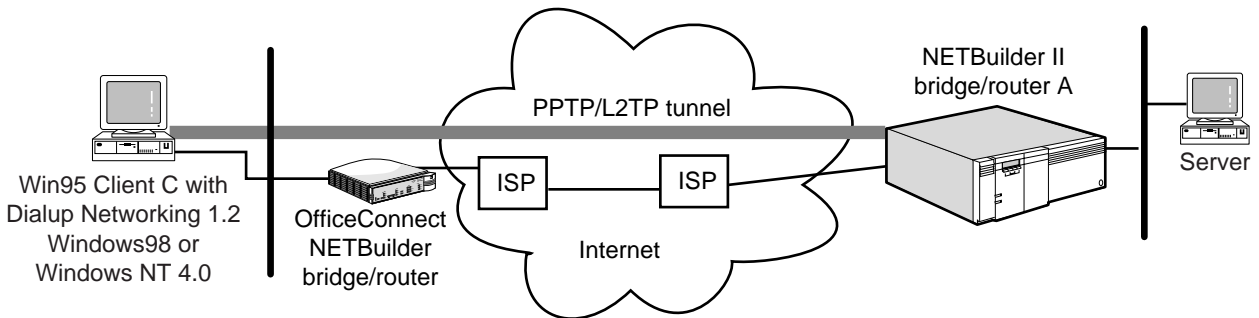


*When using this type of client, you do not need to enable the MLP service for the remote clients. The other steps for setting up this configuration are the same as those in the procedure, except you do not need to configure the L2Tunnel AccessList parameter.*

### Internet Based Remote Access

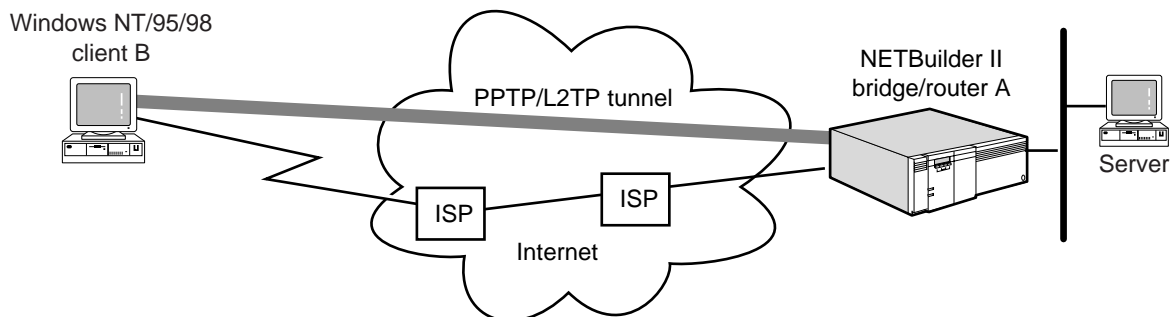
The configuration shown in Figure 13 is similar to the one shown in Figure 12, except that client C has a direct Internet connection using an OfficeConnect NETBuilder bridge/router. To gain the enterprise access, a PPTP tunnel is created between client C and the NETBuilder RAS server.

The OfficeConnect bridge/router is directly connected to the Internet via an ISP connection where the bridge/router acts as the default gateway for the remote client. In this configuration, you do not need to enable the MLP service for the remote clients. The other steps for setting up this configuration are the same as those in the procedure. Remember, the OfficeConnect bridge/router must be configured in the L2Tunnel AccessList.

**Figure 134** Internet Based Remote Access

### OfficeConnect NETBuilder with NAT at the Remote Office

In the configuration shown in Figure 14, an OfficeConnect NETBuilder bridge/router is configured using the QuickStep VPN application and is connected to the Internet via an ISP connection. Once connected, the OfficeConnect bridge/router establishes a PPTP tunnel to the enterprise NETBuilder bridge/router A. All remote stations attached to the OfficeConnect bridge/router have access to the enterprise network. The PPTP connection is transparent to the remote stations. For more information about configuring the PPTP tunnel, see the Configuring L2Tunnel Connections chapter.

**Figure 135** OfficeConnect NETBuilder with NAT

The OfficeConnect NETBuilder bridge/router at the remote site is configured to run Network Address Translation (NAT). The central site NETBuilder II bridge/routers view the OfficeConnect NETBuilder bridge/router as a RAS client. However, the OfficeConnect bridge/router is the default router for the remote Windows clients and performs NAT for all the remote stations using a private IP network at the remote site.

## Configuring NCPs

When the NETBuilder bridge/router establishes a Link Control Protocol (LCP) connection with a RAS client, it sends out the following Network Control Protocol (NCP) Configuration request packets:

- IPCP
- SNA-802.2
- SNA
- OSI

- XNS
- IPXCP

These request packets are sent out regardless of whether the RAS client supports these NCPs or not. In the 11.2 release, Enterprise OS software implements a scheme where the NCPs are configurable. Only the configured NCPs are negotiated. IPX is not part of the NCP configuration list since it is controlled by the IPX RAS component.

These enhancements are part of the RAS support, and are available on the NETBuilder II, Super Stack II, and OfficeConnect NETBuilder platforms.

The NCP configuration list is in the PPP component and is system-wide. You can enable/disable an NCP by specifying the appropriate element in the list. If an NCP is disabled, the NETBuilder bridge/router does not send out a configuration request packet for that NCP and ignores the incoming configuration request for that NCP.

You can apply the SETDefault command to modify the list and SHow to view the list. Since this command is system-wide, the command "SHowD -PPP CONF" does not apply. The default value is IPCP enabled, the rest disabled.

**Example** To specify the list of negotiable NCPs, follow these steps:

- 1 To set up the NCP configuration list, enter:

```
SETDefault -PPP NcpProtocolId = (IPcp, Osi, Xns, SNa-802.2, Sna)
```

- 2 To display the list, enter:

```
SHow -PPP NcpProtocolId
```

The default is npi = (IPcp, NoOsi, NoXns, NoSna-802.2, NoSna).

---

## RADIUS Framed Net Mask and Framed Route Attributes

Before the Enterprise OS software version 11.2, RAS only supported RAS clients (for example, hosts). With the implementation of the RADIUS framed net mask and framed route, the RAS client can be a router that connects its network to the NETBuilder II networks. An IP host on the NETBuilder II bridge/router LAN can now talk to an IP host on the RAS router's LAN.

For a RAS client to be considered as a router, both the framed IP address and the framed net mask attributes must be configured on the RADIUS server.

**Example** To configure the RADIUS server, follow these steps:

- 1 To add a static route entry to the routing table, enter:

```
Dest. addr = Framed IP Addr & Framed Net Mask (ie network number)
Gateway = !virtual Port of this RAS router
Metric = 0
Owner = STATIC
```

This routing entry is not saved in CCS and is deleted when the RAS connection is torn down.

For each framed route entry defined in the RADIUS server, a static route entry is added to the routing table with the following information

```
Dest. addr = Framed Route's dest. addr & Framed Route's net mask
Gateway = !virtual Port of this RAS router
Metric = Framed Route's Metric
Owner = STATIC
```

Since all packets destined for the framed route's destination address must go through the RAS virtual port, the framed route's gateway is ignored, and is replaced by the RAS virtual port.

The framed route format must follow the RFC 2138 recommendation, for example:

```
140.1.2.0 255.255.255.0 140.1.2.1 2, or
140.1.2.0/24 140.1.2.1 2
```

This routing entry is not saved in CCS and is deleted when the RAS connection is torn down.

The RAS IP address is not added to the ARP table as in the RAS Client case since it is not a host. When the command "sh -ras Connection" is issued, the RAS IP address has the format "Framed IP addr/Framed Net Mask (# of netmask bits)", for example, 140.1.2.160/24.

#### **RADIUS NAS-Port-Type Attribute**

The RADIUS NAS-Port-Type attribute indicates the type of NAS physical port that is authenticating the user. It can have the following values:

- 0 Async
- 1 Sync
- 2 ISDN Sync
- 3 ISDN Async V.120
- 4 ISDN Async C.110

When the RAS component receives a login request from a remote client, it extracts this information from the PPTP port CB and sends it in the RADIUS Access Request to the RADIUS server. Similarly, when the remote client closes down the RAS connection, this information is added to the accounting stop packet to be sent to the RADIUS server.

---

#### **Logging Messages**

RAS can be configured to log system messages to the AuditLog Service, to the local console, or both. Log messages contain crucial information such as the date and time, interface, incoming and outgoing, packet header summary, and reason. Each message contains two types of codes: facility and priority. The facility code tells syslog what subsystem the message is from and the priority code tells syslog how important the message is (ranging from Log(0), emergency, which is the highest priority to Log(7), debug, which is the lowest priority).



To log messages to the AuditLog Service, enable the service using:

```
SETDefault -AuditLog CONTROL = (CONFIG, MESSAGES, SECURITY)
```

To specify whether messages are logged to the AuditLog Service, the local console, or both, use:

```
SETDefault -RAS Log = ([Syslog | NoSyslog], [Console | NoConsole],
[Connect | NoConnect], [AuthFail | NoAuthFail], [RsrcFail | NoRsrcFail
])
```

### Specifying Log Content

You can enable specific information to be logged. For example, to generate a log message when a session is disconnected, enter:

```
SETDefault -RAS Log = NoConnect
```

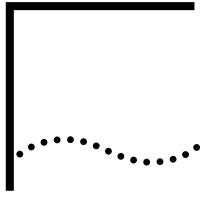
If you want to log authentication failures, enter:

```
SETDefault -RAS Log = AuthFail
```

If you want to log connections which fail due to lack of resources, enter:

```
SETDefault -RAS Log = RsrcFail
```





# IP SECURITY OPTIONS

This chapter describes how to configure your bridge/router to implement IP security options to protect datagrams at specified classification levels under the protection rules of specific authorities. These security features, which comply with RFC 1108, are necessary for any network implementing IP security options; for example, Department of Defense networks.

Your system can use Internet Protocol (IP) security options in an internetwork to:

- Transmit the common security labels from source to destination.
- Ensure that the route taken by the datagram is protected to the level required by all protection authorities indicated in the datagram.

This chapter also describes the use of source IP spoofing as a common type of security violation and provides the Internet Computer Emergency Response Team (CERT) recommendations for preventing this type of network attack.



*For conceptual information, see "IP Security Terms" later in this chapter.*

---

## Configuring IP Security Parameters for End Systems

This section describes how to configure IP security parameters for end system configurations (IP routing is not being used). Parameters related to IP security that may need to be configured depend on the type of networking devices involved and the amount of security required.

The procedure is an example of how to use the IP security option commands and is not a standard configuration procedure. Depending on the type of network security you require, your configuration procedure may differ from the one provided.

### Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

If you are using your system as an end system (the system receives packets and sends packets through the same interface; packets are not routed through the box to another interface), a configuration example is provided in the next section. If you are using your system as an IP router, a configuration example is provided in "Configuring IP Security Options for IP Routers" later in this chapter.

Before beginning the configuration, make sure your system has the following initial settings for the IP security option parameters:

- SecLEVel is set to UNCLass UNCLass for all ports.

- SecAuthIn is set to GENSER for all ports.
- SecCONTROL is set to NoEXTended for all ports.
- SecLabelSys is set to UNClass GENSER for all ports.
- SecLabelValues is set to RFC1108.

The following procedure describes how to configure your system to transmit and receive datagrams with a TopSECRET classification level, how to accept datagrams with any combination of GENSER and SIOP-ESI protection authorities, and how to attach a TopSECRET GENSER label to datagrams originated by the system. The IP address was assigned to the system using the SETDefault !0 -IP NETaddr command.

**Procedure** To configure the system, follow these steps:

- 1 To configure all system ports to transmit and receive TopSECRET datagrams enter:

```
SETDefault !0 -IP SecLEVEL = TopSECRET
```

- 2 Specify the protection authorities that can be present in datagrams received on all ports.

- a Change the default setting by entering:

```
DElete !0 -IP SecAuthIn GENSER
```

- b Set the Security Authorization SIOP-ESI protection by entering:

```
ADD !0 -IP SecAuthIn GENSER SIOP ANY
```

- 3 Configure the classification level and protection authority label for datagrams originated by the system by entering:

```
SETDefault !0 -IP SecLabelSys = TopSECRET GENSER
```

- 4 Enable the system to perform security processing of packets received from a file server by entering:

```
SETDefault -IP SecFileServer = Yes
```

The default for the SecFileServer parameter is "No." For the system to communicate with the file server when IP security options are enabled, you must set the SecFileServer parameter to "Yes."

For more information, see the SecFileServer parameter in the IP Service Parameters chapter in *Reference for Enterprise OS Software*.

- 5 Enable security options on the system by entering:

```
SETDefault -IP CONTROL = SECURITY
```

For information on how to check your configuration, see "Verifying IP Security Options" later in this chapter.

---

## Configuring IP Security Options for IP Routers

This section describes how to configure IP security parameters for IP router configurations. Parameters related to IP security that may need to be configured depend on the type of networking devices involved and the amount of security required.

The procedures are an example of how to use the IP security option commands and are not standard configuration procedures. Depending on the type of network security you require, your configuration procedure may differ from the one provided.

- Prerequisites** Before beginning the configuration, make sure your system has the following initial settings for the IP security option parameters:
- SecLEVel is set to UNCLass UNCLass for all ports.
  - SecAuthIn and SecAuthOut are set to GENSER for all ports.
  - SecLabelDefault is set to NONE for all ports.
  - SecCONTRol is set to NoEXTended, NoBasicFirst, NoLabelAdd, and NoLabelStrip for all ports.
  - SecLabelSys is set to UNCLass GENSER for all ports.
  - SecLabelValues is set to RFC 1108.

**Procedures** Figure 136 is an example of a typical internetwork in which IP security options are configured. The configuration allows the following communications:

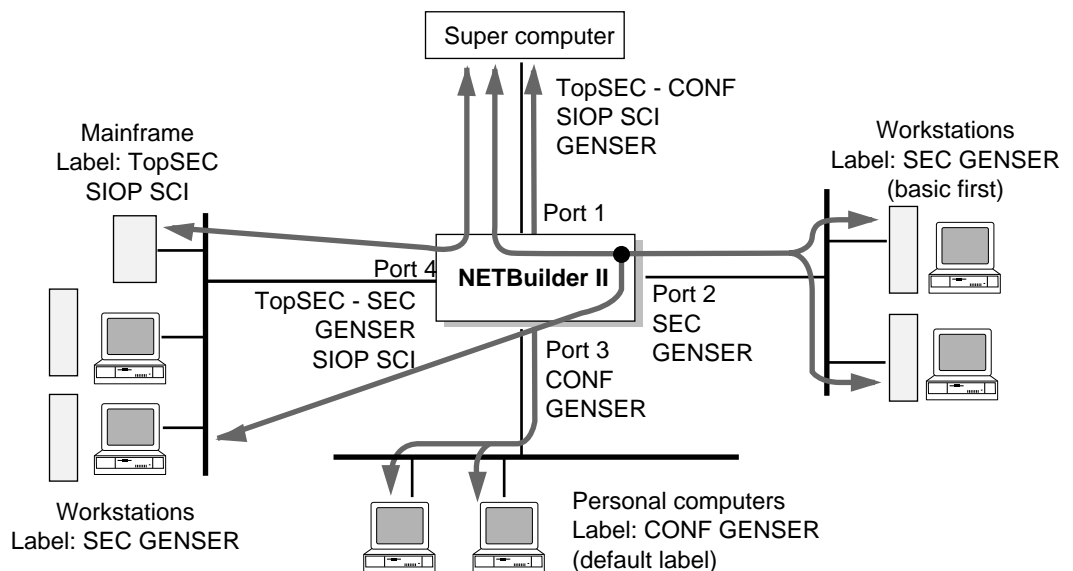
- PCs with the supercomputer
- Workstations with other workstations as well as the supercomputer
- Mainframes with the supercomputer

In Figure 136, the devices are performing the following operations:

- Mainframes generate the label "TSEC SIOP SCI."
- Workstations on port 2 and 4 generate the label "SEC GENSER" and require the basic option to be first in the IP header.
- PCs can neither generate nor receive labels. A default label of "CONF GENSER" is generated for them. This label is stripped before a datagram is sent on port 3.
- Supercomputer assigns labels based on host addressing. It needs to generate datagrams with labels "SEC GENSER" when communicating with workstations, "CONF GENSER" for PCs, and "TSEC SIOP SCI" for mainframes.

A description of each port interface is provided in addition to examples of how to set the IP security option parameters.

**Figure 136** IP Security Options Configuration Example



## Port 1 Configuration

This procedure shows how to configure port 1 of the system based on Figure 136. Port 1 connects to a supercomputer, which assigns labels based on host addressing. You want to configure the system to allow the supercomputer to communicate with the workstations on ports 2 and 4, the PCs, and the mainframes. You need to configure port 1 to transmit to and receive from the supercomputer datagrams with a TopSEcRet, SEcRet, or CONFidential classification level and with protection authority flags SIOP-ESI and SCI both set, or just GENSER set.

To configure port 1, follow these steps:

- 1 Specify the range of security levels of the datagrams that can be transmitted and received on port 1 by entering:

```
SETDefault !1 -IP SecLEVel = CONFidential TopSEcRet
```

The system can receive or transmit datagrams from port 1 with classification levels of CONFidential, SEcRet, or TopSEcRet.

- 2 Specify the protection authorities that can be present in datagrams received on port 1 by entering:

```
ADD !1 -IP SecAuthIn SIOP SCI
```

The system can receive datagrams from the network on port 1 with SIOP-ESI and SCI set, or just GENSER set. GENSER appears in the SecAuthIn table by default.

- 3 Specify the protection authorities that can be present in datagrams transmitted on port 1 by entering:

```
ADD !1 -IP SecAuthOut SIOP SCI
```

The system can transmit datagrams to the network on port 1 with SIOP-ESI and SCI set, or just GENSER set.

- 4 Configure a single classification level and protection authority label for datagrams originated by the system and transmitted on port 1 by entering:

```
SETDefault !1 -IP SecLabelSys = CONFidential GENSER
```

Any datagram generated by the system, including Internet Control Message Protocol (ICMP) messages, have this label when transmitted on port 1.

- 5 Configure the system so that a label is attached to datagrams before transmission over port 1 by entering:

```
SETDefault !1 -IP SecCONTrol = LabelAdd
```

This parameter is configured because PCs cannot generate labels. The system must be configured to attach a label to a datagram destined for the supercomputer. The label that is attached to the datagram before transmission to the supercomputer is based on the value of the SecLabelDefault parameter. This parameter is set on port 3 of the system.

## Port 2 Configuration

This procedure shows how to configure port 2 of the system based on Figure 136. Port 2 connects to workstations, which require the basic security option to be the first option in the IP header. You want to configure the system to allow the workstations to communicate with the supercomputer and the workstations on port 4. You need to configure port 2 to transmit to and receive from the workstations datagrams with a SEcRet classification level and with the GENSER

protection authority flag. You also need to configure the port so that the basic security option is the first option in the IP header of datagrams transmitted on this port.

To configure port 2, follow these steps:

- 1 Specify the security level of datagrams that can be transmitted and received on port 2 by entering:

```
SETDefault !2 -IP SecLEVel = SECRet
```

The system can receive or transmit datagrams from port 2 with classification level of SECRet.

- 2 Configure port 2 so that the basic security option is the first option in the IP header of datagrams transmitted on port 2 by entering:

```
SETDefault !2 -IP SecCONTRol = BasicFirst
```

For datagrams transmitted on port 2, the workstations require that the basic security option is the first option in the IP header.

- 3 Configure a single classification level and protection authority label for datagrams originated by the system and transmitted on port 2 by entering:

```
SETDefault !2 -IP SecLabelSys = SECRet GENSER
```

Any datagram generated by the system, including ICMP messages, have this label when transmitted over port 2.

### Port 3 Configuration

This procedure shows how to configure port 3 of the system based on Figure 136. Port 3 connects to PCs. You want to configure the system to allow the PCs to communicate only with the supercomputer. You need to configure port 3 to transmit and receive datagrams with a CONFidential classification level and with GENSER protection authority flag. Because the PCs can neither generate nor accept a security label, the system must attach a default label (CONFidential GENSER) to datagrams received on port 3 and destined for the supercomputer, and strip the label from datagrams destined for the PCs.

To configure port 3, follow these steps:

- 1 Specify the security level of datagrams that can be transmitted and received on port 3 by entering:

```
SETDefault !3 -IP SecLEVel = CONFidential
```

The system can receive or transmit datagrams by this port with classification level of CONFidential.

- 2 Configure port 3 to strip the security label from datagrams transmitted to the PCs by entering:

```
SETDefault !3 -IP SecCONTRol = LabelStrip
```

Because PCs cannot receive labels in datagrams, the system must strip the label before transmission on port 3.

- 3 Configure port 3 to attach a default label to datagrams received from PCs by entering:

```
SETDefault !3 -IP SecLabelDefault = CONFidential GENSER
```

PCs cannot generate or transmit labels; therefore, the system must attach a default label of CONFidential and GENSER. The datagram can then be properly routed to port 1 and the supercomputer.

- 4 Configure port 3 so that datagrams originated by the system and transmitted on port 3 do not have labels by entering:

```
SETDefault !3 -IP SecLabelSys = NONE
```

Because PCs cannot receive labels, the SecLabelSys parameter needs to be set to NONE.

#### Port 4 Configuration

This procedure shows how to configure port 4 of the system based on Figure 136. Port 4 connects to mainframes and workstations. You want to configure the system to allow the mainframes to communicate only with the supercomputer, and the workstations to communicate both with the supercomputer and the workstations on port 2. You need to configure port 4 to transmit and receive datagrams with a TopSECRet or SECRet classification level, and with SIOP-ESI and SCI protection authorities set or GENSER set.

To configure port 4, follow these steps:

- 1 Specify the security level of datagrams that can be transmitted and received on port 4 by entering:

```
SETDefault !4 -IP SecLEVel = SECRet TopSECRet
```

The system can receive or transmit datagrams on this port with classification levels of SECRet and TopSECRet.

- 2 Specify the protection authorities that can be present in datagrams received on port 4 by entering:

```
ADD !4 -IP SecAuthIn SIOP SCI
```

The system can receive datagrams from the network on port 4 with SIOP-ESI and SCI set, or just GENSER. GENSER appears in the SecAuthIn table by default.

- 3 Specify the protection authorities that can be present in datagrams transmitted on port 4 by entering:

```
ADD !4 -IP SecAuthOut SIOP SCI
```

To transmit datagrams to the mainframes and workstations connected to port 4, datagrams must have the SIOP-ESI and SCI protection authority flags set in the security label of the datagram. Using this protection authority, the system can receive datagrams from the supercomputer and route them to the mainframes and workstations on port 4. With the GENSER authority (the default), the system can transmit datagrams between the supercomputer and workstations on ports 2 and 4.

- 4 Configure a single classification level and protection authority label for datagrams originated by the system and transmitted on port 4 by entering:

```
SETDefault !4 -IP SecLabelSys = SECRet GENSER
```

Any datagram generated by the system, including ICMP messages, have this label when transmitted on port 4.



## Enabling IP Security Processing

To enable the IP router security options, follow these steps:

- 1 Enable security options on the system by entering:

```
SETDefault -IP CONTROL = SECURITY
```

After enabling security options, change the default of the SecFileServer parameter to Yes to ensure proper communication with the file server. For more information, see the SecFileServer parameter in the IP Service Parameters chapter in *Reference for Enterprise OS Software*.

- 2 Display the configuration settings by entering:

```
SHOW -IP CONFIGURATION
```

For information on how to check your configuration, see “Verifying IP Security Options” later in this chapter.

## Configuring Extended Security Option Labels

For environments requiring extra security measures, you can add extended security labels to IP packets leaving specific ports. With this option, you can add a string to outgoing IP packets so that only specific hosts can accept the packets. The extended security label options can be used with the normal IP security options described earlier in this chapter, or they can be used independently.

The extended security label option is not required for the majority of configurations. If you use this option, be careful to configure it correctly to obtain the desired effect.

To support this configuration, follow these steps:

- 1 If you have basic IP security control enabled, make sure the SecCONTROL parameter is set to EXTended and NoBasicFirst for port 2 by entering:

```
SETDefault !2 -IP SecCONTROL = (EXTended, NoBasicFirst)
```

If you do not specify these values, the extended labels will be discarded.

- 2 Specify the extended security label to be added to packets using:

```
SETDefault -IP SecLabelXtra = "<string>"
```

Specify the string as values of individual bytes given as a decimal number, with each byte being separated by a slash (/). The total number of bytes must be a multiple of four, and the string must end with a slash. No syntax checking is performed, therefore, the string must be specified correctly.

- 3 Configure port 2 to add the label specified by the SecLabelXtra parameter to all packets sent over this port by entering:

```
SETDefault !2 -IP SecCONTROL = LabelXtraAdd
```

- 4 Configure port 1 to strip the extended label for all packets sent over this port by entering:

```
SETDefault !1 -IP SecCONTROL = LabelExtStrip
```

For more information about the parameters described in this procedure, see the IP Security Options chapter in *Reference for Enterprise OS Software*.

---

## Verifying IP Security Options

To check your configuration before using the IP security settings on your system, follow these steps:

- 1 Verify the IP security configuration by entering:

**SHow -IP CONFIguration**

The system displays all IP configuration settings. If any of the IP security settings are incorrect, you can reconfigure them.

- 2 Check the IP security configuration by entering:

**SecCheck**

Because no port number is specified, the system checks all port configurations. If you specify a port number with this command, the system checks the configuration for that port.

This command is a diagnostic tool and does not check for all possible configurations. If the system finds misconfigurations, warning messages are displayed.

---

## ICMP Error Messages

The following ICMP error messages may be generated as a result of security processing:

- ICMP Parameter Problem Missing Option, Type 12 Code 1, Pointer =130  
No security options in packet, but security options required on the port.
- ICMP Parameter Problem Missing Option, Type 12 Code 1, Pointer = malformed option  
Malformed security option, for example, an invalid label.
- ICMP Destination Unreachable Communication Administratively Prohibited, Type 3  
Code 9 for the network, Code 10 for the host. Security label out of range.

These ICMP error messages may be generated because of an incorrect configuration or may indicate an unauthorized break into a secure network. The network manager can decipher ICMP message codes by using a network analyzer.

To display a count of ICMP error messages, enter:

**SHow -SYS STATistics -IP**

For a sample display and explanation of the entries, see the Statistics Displays appendix.

---

## Preventing Security Attacks on IP Routers

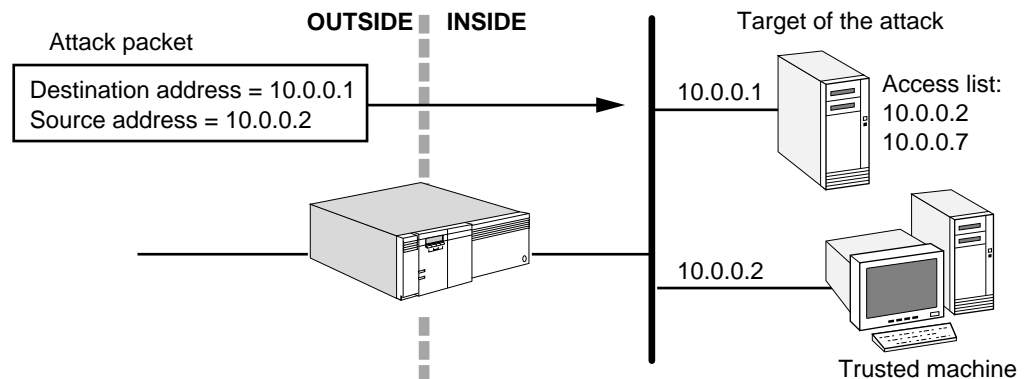
Source IP spoofing is a common type of security violation. The Internet CERT has summarized the danger of how IP spoofing is used in network attacks. This section describes how to configure 3Com bridge/router software to provide security against this type of attack.

### How IP Spoofing Works

To generate an attack, the intruder creates packets with spoofed source IP addresses. In this type of attack, the intruder transmits packets from outside the "protected" domain that claim to be from a trusted machine inside the "protected" domain (for example, the packet contains the source IP address of a

trusted machine). If the router is not configured to filter incoming packets whose source address is in the local domain, it forwards the traffic and the targeted system may become compromised. A router generally forwards this traffic because it only examines the destination IP address when it makes its forwarding decision, not the source IP address. Figure 137 illustrates the operation of a spoofed source IP address attack.

**Figure 137** Spoofed Source IP Address Attack



Attacks are aimed at applications that use authentication based on source IP addresses. If successful, an attack leads to unauthorized user and possibly root access on the targeted system. It is important to note that the described attack is possible even if no reply packets can reach the attacker. Also, disabling source routing at the router does not provide protection from this type of attack.

Examples of configurations that are potentially vulnerable to attack include:

- Routers to external networks that support multiple internal interfaces
- Routers with two interfaces that support subnetting on the internal network
- Proxy firewalls where the proxy applications use the source IP address for authentication

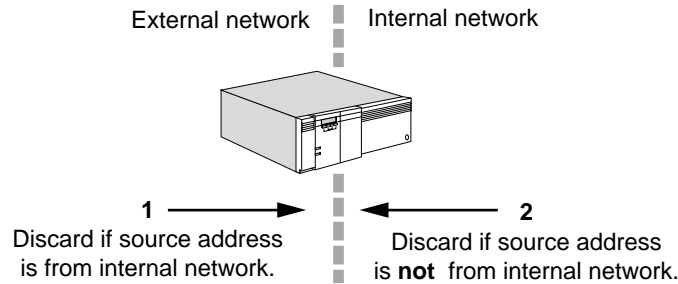
**Hijacking Tool** After intruders have achieved root access on a system, they use a tool to dynamically modify the UNIX kernel. This modification allows them to hijack existing terminal and logon connections from any user on the system. In taking over existing connections, intruders can bypass one-time passwords and other strong authentication schemes by tapping the connection after the authentication is complete. For example, a legitimate user may connect to a remote site through a logon or terminal session. An intruder can hijack the connection after the user has completed authentication to the remote location. The site would now be compromised. Currently, the hijacking tool is used primarily on SunOS 4.1.x systems. However, system features that make this attack possible are not unique to SunOS.

**Preventing Attacks** To prevent this type of attack, the CERT Coordination Center recommends that network security personnel follow these steps:

- 1 Install a filtering router that restricts the input to the external interface (known as an input filter) by not allowing a packet through if it has a source address from the internal network.

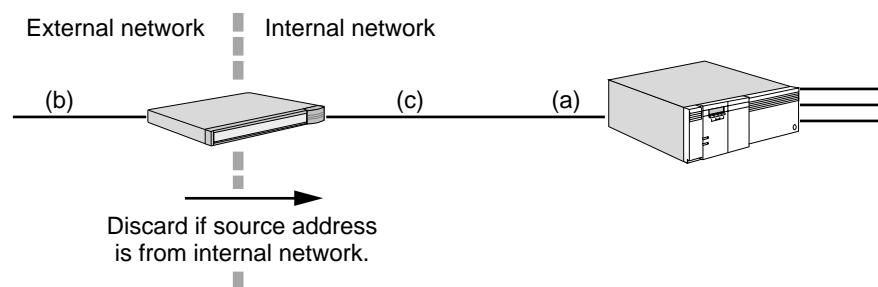
- 2 Filter outgoing packets that have a source address different from the internal network to prevent an attack originating from the local site. Figure 138 illustrates the CERT recommendations.

**Figure 138** CERT Recommended Filters



CERT recommends an alternative solution if a router does not support filtering on the in-bound side. The spoofed IP packets may be filtered by installing a second router between the original external interface (a) and the outside connection (b). This router can then be configured to block all packets that have a source address in the internal network on the outgoing interface (c) connected to the original router. Figure 139 illustrates the alternative CERT recommendation.

**Figure 139** Alternative CERT Configuration



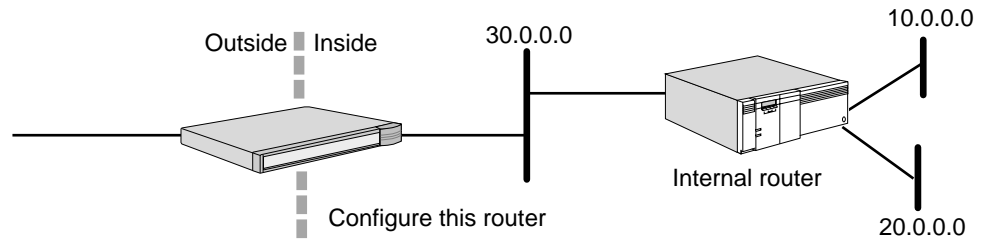
### Secure Configuration Solutions

The following examples illustrate how bridge/router software can be configured to support the CERT Advisory recommendations. Each of these examples assumes that the value of the `-IP FilterDefAction` parameter is configured to `Forward`. However, none of these examples prevent a source IP spoofing attack originating from the local site.

### Noncontiguous IP Networks

The example in Figure 140 illustrates a two-router solution where the internal network is configured with noncontiguous IP network numbers. The filters are installed on the border router, which can only have two interfaces. In a two-port router, an output filter on one port is equivalent to an input filter on the other port.

Figure 140 Noncontiguous IP Networks

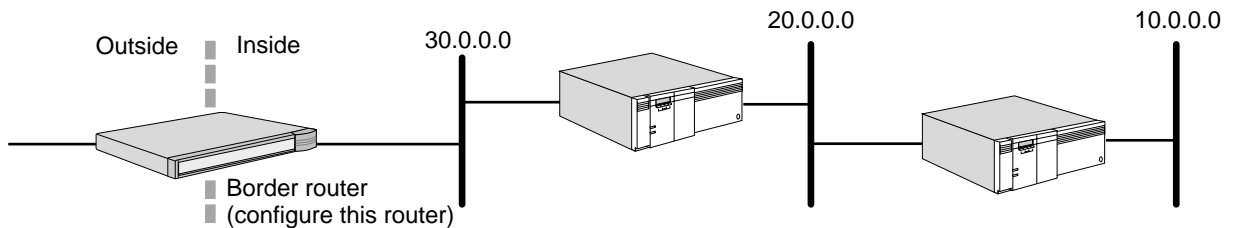


Add the following filters to the border router to prevent an external attack:

```
ADD -IP FilterAddrs 10.0.0.0/0.255.255.255 > 10.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 20.0.0.0/0.255.255.255 > 20.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 30.0.0.0/0.255.255.255 > 30.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 10.0.0.0/0.255.255.255 <> 20.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 10.0.0.0/0.255.255.255 <> 30.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 20.0.0.0/0.255.255.255 <> 30.0.0.0/0.255.255.255 Discard
```

This configuration prevents the external attack and allows the internal router to route traffic between networks 10.0.0.0, 20.0.0.0, and 30.0.0.0. This configuration also works for the cascade topology shown in Figure 141.

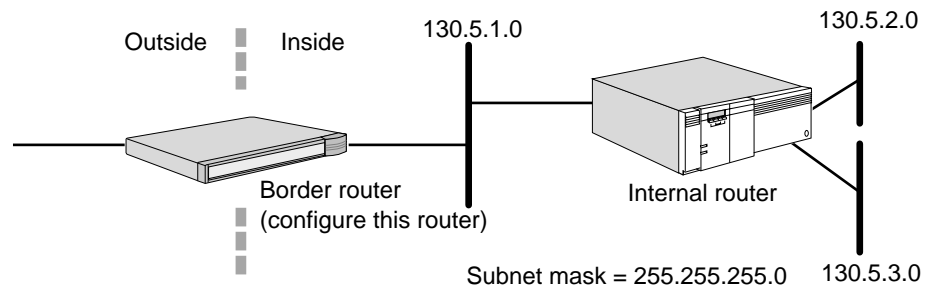
Figure 141 Noncontiguous IP Networks (Alternative Topology)



### Subnets on the Internal Network

The example in Figure 142 illustrates a two-router solution when the internal network is configured with multiple subnets of the Class B network address, 130.5.0.0.

Figure 142 Subnets on the Internal Network



Add the following filter to the border router to prevent an external attack:

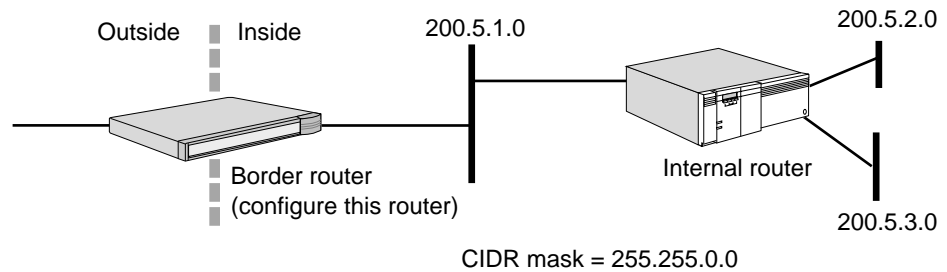
```
ADD -IP FilterAddrs 130.5.0.0/0.0.255.255 > 130.5.0.0/0.0.255.255 Discard
```

This configuration prevents the external attack and allows the internal router to route traffic between all subnetworks of 130.5.0.0. In this example, a single filter can protect multiple subnets.

### Multiple Contiguous IP Networks

The example in Figure 143 illustrates a two-router solution where the internal network is configured with contiguous IP network numbers. Assume the service provider has provided the subscriber with the Classless Interdomain Routing (CIDR) Protocol block 200.5.0.0/255.255.0.0.

**Figure 143** Multiple Contiguous IP Networks



Add the following filter to the border router to prevent an external attack:

```
ADD -IP FilterAddrs 200.5.0.0/0.0.255.255 > 200.5.0.0/0.0.255.255 Discard
```

This configuration prevents the external attack and allows the internal router to route traffic between supernets of 200.5.0.0/255.255.0.0. In this example, a single filter can protect multiple contiguous IP networks numbers assigned as a CIDR block.

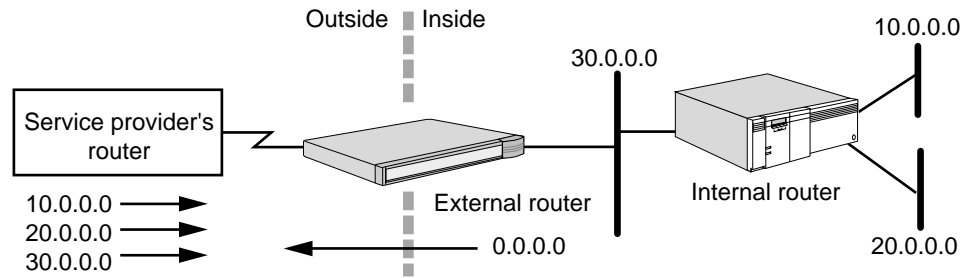
### Alternative Two-Router Configurations

Various 3Com bridge/routers can be configured for security. The external router can be a model 227 or 228 SuperStack II NETBuilder bridge/router while the inside router can be another 3Com router. In some cases, routers from two different vendors may be optimal because a bug or back door that allows entry by a hacker in one vendor's code may not exist in the other vendor's code.

In many cases, the network topology can have the following characteristics:

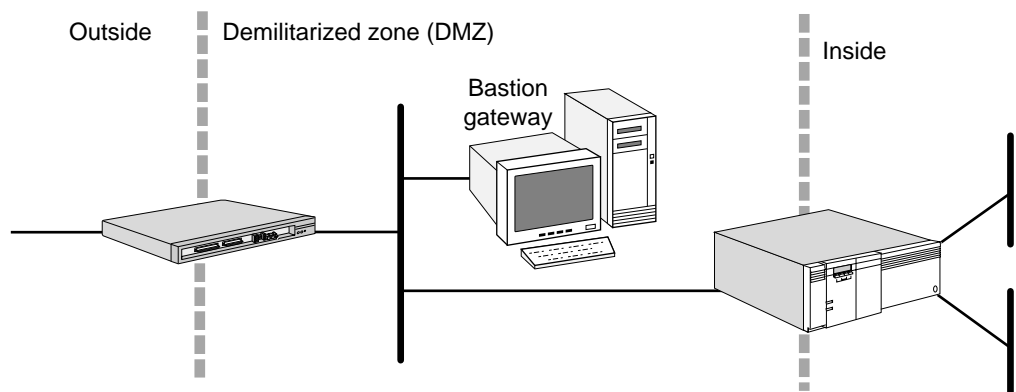
- An external link to the Internet, which is a simple serial link to the network provider's router.
- The inside network consists of a few noncontiguous networks or subnets of a single network number.

Figure 144 illustrates this common configuration. The external router is configured with the required filters. The external router is also configured with a default route pointing to the Internet. The service provider installs static routes in their router that point to the customer's network. For this configuration, it is not necessary to run a routing protocol over the external link. If the network connectivity is more complex and you are connected using a multipoint technology such as X.25 or Frame Relay, you can run the Border Gateway Protocol version 4 (BGP-4) on a model 227 or 228 SuperStack II NETBuilder bridge/router to provide the required connectivity.

**Figure 144** Two-Router Configuration

## Firewall Configurations

Many firewall configurations require the use of two routers. A typical Internet firewall using two routers is illustrated in Figure 145.

**Figure 145** Internet Firewall

In this example, the routers create a packet filtering firewall while the bastion gateway functions as an application gateway firewall. In addition to using routers, creating a secure Internet firewall requires packet filtering and applications gateways. For information about filtering, see the Configuring Mnemonic Filtering chapter.

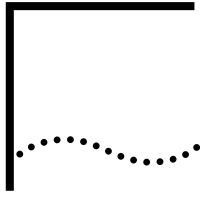
## IP Security Terms

The following terms used in this chapter explain IP security concepts:

basic security options	Identifies the U.S. classification level at which the datagram is to be protected and the authorities whose protection rules apply to each datagram.
classification level	Specifies the U.S. classification level (top secret, secret, confidential, and unclassified) at which the datagram must be protected.
Classless Interdomain Routing (CIDR)	A method of using IP addresses without regard to traditional address classes to help solve the problem of the lack of class B network numbers.
extended security options	Permits additional security labeling information beyond what is presented in the basic security option to meet the needs of additional registered authorities.

label	Refers to the classification level and protection authority characteristics of a datagram.
protection authority	<p>Identifies the agency that specifies the protection rules for transmission and processing of information contained in the datagram. Examples of protection authorities include the following:</p> <p>GENSER: the point of contact for this authority is the Designated Approving Authority per Department of Defense (DOD) 5200.29.</p> <p>SLOP-ESI: The point of contact for this authority is the Department of Defense, Organization of the Joint Chiefs of Staff.</p> <p>SCI: The point of contact for this authority is the Director of Central Intelligence.</p> <p>NSA: The point of contact for this authority is the National Security Agency.</p> <p>DOE: The point of contact for this authority is the Department of Energy.</p>
source IP spoofing	A common type of security violation in which an intruder accesses a protected domain by using the IP address of a trusted machine.





# CONFIGURING IPSEC

This chapter describes how to configure the IP Security Protocol (IPSec) on your IP router. IPSec provides security at the network layer. Because IPSec is integrated into IP itself, IPSec adds security to any link, regardless of the application used.

IPSec can be used in conjunction with a tunneling protocol. The protocols that can be used for tunneling are: PPTP, L2TP, IPIP. See the Configuring L2Tunnel Connections chapter for more information about PPTP/L2TP.



*It is recommended that IPSec control or the PORT service control be disabled while configuring policies and enabled only after all IPSec policy configuration has been completed.*

For conceptual information, see “How IPSec Works” later in this chapter.

---

## Configuring IPSec

The procedures in this section describe how to define the basic components of IPSec. IPSec can be configured using manual policies and keys or using dynamic policies. Also, it can be configured in either transport mode or tunnel mode.

**Transport Mode** Transport mode security associations are used to protect traffic that is viewed on an end system from an IPSec perspective. For example, it can be used with PPTP/L2TP/IPIP tunnels or to serve network management traffic like Telnet or SNMP.

**Tunnel Mode** Tunnel mode security associations are used to protect IP traffic forwarded by the router on IPSec tunnel ports.

## Creating Manual Policies

An IPSec policy consists of an action, the packet types that require the action, and the source and destination addresses between which the action occurs. The following seven actions are supported:

- Action AhXport
- Action EspXport
- Action AhEspXport
- Action EspAuthXport
- Action AhTunnel
- Action EspAuthTunnel
- Action EspTunnel

## Configuring Manual Security Policies

To configure IPsec using manual policies, follow these steps:

- 1 Define a policy using ManualPolicy.
- 2 Define the manual keys using the KeySet parameter.
- 3 Bind the information together using the ManualKeyInfo parameter.

To configure a manual security policy, use:

```
ADD !<portlist> manualPOLICY <policy_name> <action> (Default |
 {<filters> <src_ipaddr/mask>
 (<dst_ipaddr/mask> | DYNAMIC)})
 [<encrypt_alg>] [<auth_alg>]

<action>: AhEspXport | AhTunnel | AhXport |
 EspAuthTunnel | EspAuthXport | EspTunnel | EspXport
<filters>: ANY | (GRE, ICMP, OSPF,
 TCP[(<port>,<port>)...up to 16 pairs],
 UDP[(<port>,<port>)...up to 16 pairs])
<encrypt_alg>: 3DES | 3DES2key | DES | RC5 | NULL
<auth_alg>: MD5 | SHA
<port>: 165535 | * | Archie | DNS | Finger | FTP | FTPData |
 Gopher | HTTP | NFS | NNTP | NTP | POP2 | POP3 |
 PortMap | RIP | SMTP | SNMP | SNMPtrap | Syslog |
 Telnet | WAIS
```

The default for encrypt\_algorithms is DES. The default for auth\_algorithms is MD5.

## Creating Key Sets

To create a key set, use:

```
ADD -IPSEC KeySet <key_set_name> [EncryptKey ("<encrypt_key>" |
"%<encrypt_key>")] [AuthKey ("<auth_key>" | "%<auth_key>")]
```



*The encrypt\_key and auth\_key must match the values on the peer system at the other end of the security association.*

<key\_set\_name> is a name you assign to the key set you are adding.

<encrypt\_key> and <auth\_key> can be 1 to 128 bytes entered as either ASCII text strings or as a series of hexadecimal digits. See "Configuring Manual Key Information" next for more information about key set usage.

To delete a key set, use:

```
DELEte -IPSEC KeySet [<key_set_name> | ALL]
```

For example, to create a new encryption key set, enter:

```
ADD !1 IPSEC KeySet esp_key EncryptKey "hello124"
```

To create a key set for both encryption and authentication, enter:

```
ADD !1 IPSEC KeySet espah_key EncryptKey "hello124" AuthKey "world236"
```

## Configuring Manual Key Information

The ManualKeyInfo parameter binds manual keying information to an IPsec policy. Only one ManualKeyInfo command can be applied to each policy. To configure manual key information, use:

```
SETDefault !<portlist> -IPSEC ManualKeyInfo = <policy_name>
 (<key_set_name> | NONE) [SpiEsp <spi_in> <spi_out>] [SpiAh <spi_in>
 <spi_out>]
```

A Security Parameters Index (SPI) value is used in conjunction with the destination address to identify a particular security association which represents a set of agreements between senders and receivers on a key, on an encryption or authentication algorithm, and on SPI numbers.

A key is specified using the ADD -IPSEC keyset command. It is later bound to an IPsec policy when an add IPsec policy command is entered. The key set and policy command can be used in any order. Binding takes place when the second of the two commands is issued.

When the key is entered no particular length restriction is applied. Keys can be entered as either ASCII text or hex values in the range of 1 to 128 bytes. When a key is bound, certain length restrictions are applied. The required key length depends on the Enterprise OS software package used.

All packages reject keys that are too short for their encryption transform and generate error messages. The xE packages truncate long keys to 7 bytes, and the xS packages truncate long keys to 24 bytes, with appropriate warning messages.



*For compatibility with previous software versions that did not enforce key lengths, it is possible to enter a DES key as an 8-byte hex value with the appropriate number of null characters at the end. For example, a DES key of abcd should now be entered: %6162636400000000*

To change the manual keying information, you must first delete the information using NONE as the key set name, then add the new information using SETDefault. For example, to create a security association and bind a key set to a corresponding encryption policy, enter:

```
SETDefault !1 -IPSEC ManualKeyInfo = esp_pol esp_key SpiEsp 500 501
```

To create a security association of an encryption and authentication policy, enter:

```
SETDefault !1 -IPSEC ManualKeyInfo = ah espah_pol espah_key SpiEsp 600 601
SpiAh 700 701
```

When keys are displayed using the SHow -IPSEC Keyset command, the MD5 hash of the key is displayed rather than the key itself. This allows you to compare keys for equality without exposing the actual key value. The length of the key is also displayed, since the hash is always a 32-digit hex value.

During boot, any previously configured policies and keys are bound together. The various length restrictions are applied during this binding, so that you cannot use keys that are longer than the package supports. At boot-time, binding accepts DES keys that are shorter than 8 bytes and the system generates a warning rather than an error.

## Configuring IPsec with Manual Policy

For example, to protect all TCP and UDP traffic between router 1 (170.0.0.1) and router 2 (180.0.0.1) on port 1 with an IPsec encryption policy, follow these steps:

- 1 Create an encryption policy with an unique policy name by entering:
 

```
ADD !1 -IPSEC manualPOLicy esp_pol EspXport tcp,udp 170.0.0.1 180.0.0.1
```
- 2 Create a key set and specify the encryption key by entering:
 

```
ADD -IPSEC KeySet esp_key EncrypKey "hello536"
```
- 3 Create a manual security association by binding the above policy and key set. Assuming SPIin is 500 and SPIout are 501, enter:
 

```
SETD !1 -IPSEC ManualKeyInfo = esp_pol esp_key SpiEsp 500 501
```
- 4 Finally, enable the IPsec policy by entering:
 

```
SETDefault !1 -IPSEC CONTtrol = Enable
```

## Configuring Dynamic-Key Security Policies

The DynamicPOLicy parameter adds dynamic-key IPsec policies to one or more ports. Dynamic policies provide protection for sensitive IP traffic traversing unsecured networks, such as the Internet, with a greater level of security than manual key policies. Dynamic policies specify:

- The type of IPsec security associations to establish;
- Which IP traffic to exchange on established security associations;
- How identified IP traffic is protected.

To configure IPsec using dynamic policies, follow these steps:

- 1 First define the traffic that needs to be protected by configuring SelectorList.
- 2 Define the type of protection using TransformList. (This defines how the data traffic is protected.)
- 3 Define how the IKE/ISAKMP negotiation is protected, using the IKEProfile parameter.
- 4 Define the PreSharedKey for authentication.
- 5 Bind the information together using DynamicPolicy.

To create a dynamic policy, use:

```
ADD [!<portlist>] -IPSEC DynamicPOLicy <policy_name> <priority> <mode>
<selctrlist_name> <xfrmlist_name> [<pfs>] [<lifetime>] <policy_name>:
unique name (1-15 chars) <priority>: 1-9999, 1 = highest <mode>: Tunnel |
Xport
<slctrlist_name>: name of SelectorList to match
<xfrmlist_name>: name of TransformList to use
<pfs>: GlobalPFS | NoPFS | (PFS [Group1 | Group2])
<lifetime>: GlobalLifeTime | {(1-1440m (min), 1-720h (hours), 1-
366d (days)), (1-1000kb | 1-1000mb)}
```

When a dynamic policy is created, it is given an unique name. This name is used to identify the policy in subsequent commands. The policy is also assigned an unique priority from 1 to 9999 to determine the preference between policies.



*Traffic that matches more than one policy is always secured by the policy with the lowest priority. Since dynamic policies may exist on several ports, their priority values must be unique across all of the ports on the system.*

IPSec policies can be either tunnel mode or transport (xport) mode security associations.

### **Selector Lists**

IPSec selector lists are used to determine which traffic will be secured by a given dynamic policy. The selector list specifies one or more types of traffic to include (or exclude) and is linked to the dynamic policy by its name. The selector list must be entered before the dynamic policy is added.

### **Transform Lists**

Dynamic policies allow a great variety of security transforms to be used to protect IP traffic. These transforms are specified in IPSec transform lists, which are named lists of protocol-transform combinations. Like selector lists, transform lists must be entered before the dynamic policy, and are included by name.

### **IKEProfile**

The IKEProfile parameter defines a group of settings for IPSec to use when establishing an IKE security association. The settings include authentication method, encryption algorithm, hash algorithm, and optionally the lifetime and Diffie-Hellman group to use in negotiations.

### **PreSharedKey**

The PreSharedKey parameter defines the preshared keys used when establishing IKE security associations using the preshared key authentication method. The key is associated with the peer or peers using the Phase 1 ID specified in peer\_Phase1ID. Key values can be entered as quoted ASCII text, or as a series of hexadecimal digits preceded by %.

Large networks can be configured easily by using the same key values across many routers. By specifying peer ID as an IP address with a subnet mask, all the peers falling within the subnet can share a single key. The Phase 1 ID 0.0.0.0/0 matches any IP address to facilitate a global shared key.

### **DynamicPolicy**

The DynamicPOLicy parameter adds dynamic-key IPSec policies to one or more ports. Dynamic policies specify whether to use tunnel or transport mode, which selector list to use to match IP traffic, and which transform list to use when encrypting and/or authenticating packets.

### **Customized Security Associations**

Two optional parameters are provided to customize the security associations created by dynamic policies: Perfect Forward Secrecy (PFS) and Lifetime.

Perfect Forward Secrecy (PFS) provides higher security by renegotiating a shared secret between IPSec peers each time a new key is needed. Since generating a shared secret demands intense numerical calculations (known as Diffie-Hellman), using this option may cause reduced performance during renegotiation.

Lifetime determines the amount of time elapsed and/or the amount of data protected by an IPsec security association before it expires. The lifetime can be specified in units of minutes (m), hours (h), days(d), and/or kilobytes (kb), and megabytes (mb). By default, policies use the value specified in the GlobalLifeTime parameter.

**Enabling IPsec** Enable IPsec policy checking on the port using:

```
SETDefault !<portlist> -IPSEC CONTROL = Enable
```



*You should only enable IPsec policy checking on ports that need IPsec protection. Enabling IPsec policy checking can decrease the performance of your router.*

For example, to enable IPsec on port 1, enter:

```
SETDefault !1 -IPSEC CONTROL = Enable
```

To disable IPsec on port 1, enter:

```
SETDefault !1 -IPSEC CONTROL = Disable
```

---

## How IPsec Works

IPsec integrates security directly into IP. IPsec provides three main areas of security: authentication, which validates the communicating parties; integrity, which makes sure the data has not been altered; and confidentiality, which ensures the data cannot be intercepted and viewed.

IPsec secures the underlying network layer. That way, an IPsec link is secure regardless of the application.

IPsec works with the existing Internet infrastructure using encapsulation. It secures a packet of data by encrypting it before sending it over the Internet. On the receiving end, an IPsec-compliant device decrypts the data.

The security protection can be selectively applied to various types of data traffic based on protocols, IP addresses, network addresses, applications (via TCP/UDP port addresses), and network interfaces. System-originated IP traffic (Telnet, OSPF, and RIP for example) can be protected by IPsec directly. SNA traffic can be protected by IPsec through the DLSw tunnel. Other multiprotocol traffic (IPX, AppleTalk, and DECnet for example) and forwarded IP traffic are protected by IPsec through the L2TP/PPTP tunnel. See the Configuring L2Tunnel Connections chapter for more information about PPTP/L2TP tunneling.

**Policies** IPsec policies allow you to protect various types of traffic based on protocols, IP addresses, network addresses, network interfaces, and applications (via port addresses).

## Encapsulation Security Payload

Encapsulation security payload (ESP) is used to provide data confidentiality via encryption. For outbound traffic, it encrypts the IP payload and inserts an ESP header between the IP header and the payload. For inbound traffic, it decrypts the IP payload and removes the ESP header.

DES and RC5 encryption algorithms are supported in the xE packages. DES-CBC is the Cipher Block Chaining (CBC) mode of the US Data Encryption Standard (DES), which uses an 8 byte key and operates on an eight-byte data block where the

output of each block is fed into the next block to avoid repeating the same cipher output for those blocks with the same cleartext data.

3DES has three stages as indicated by its name. These stages include an encryption stage, a decrypting stage, and another encryption stage. 3DES keys must be at least 16 bytes long for the xS packages. The 3DES key is constructed using the first and third 8 bytes for the encrypt phase, and the second 8 bytes for the decrypt phase.

Key lengths are enforced when they are entered. Warning messages inform you when the entered key does not meet the requirements.

Entered keys longer than the supported maximum length for the chosen crypto algorithm and the package are truncated as necessary.



*Encryption CANNOT be exported without a legal export license. See the release notes for your software for export restrictions.*

ESP can be applied alone or with authentication headers.

### Authentication Header

Authentication header (AH) is used to provide data integrity and data origin authentication and to provide protection against replays using the HMAC-MD5 or HMAC-SHA1 crypto algorithm. For outbound traffic, AH computes integrity checksum value (ICV) and inserts an authentication header between the IP header and the higher layer protocol header. For inbound traffic, AH verifies the ICV and removes the AH. AH can be applied alone or with ESP.

HMAC-MD5 and HMAC-SHA1 are standards-based hash algorithms. In general, HMAC-SHA1 requires more computation and is considered to be more secure but slower.

### IP Payload Compression

IP payload compression is used to reduce the size of IP datagrams. Each IP datagram is compressed and decompressed by itself, without any relation to other datagrams. The compression of output IP datagrams is performed before any IP security processing. Similarly, the decompression of inbound IP datagrams is applied after the completion of all IP security processing. IP payload compression is negotiated dynamically, and uses the LZS compression algorithm.

---

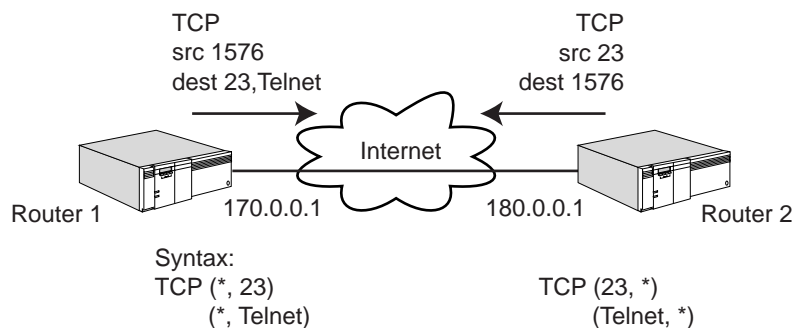
### Sample Configurations

The examples presented in this section illustrate configurations of the following topologies employing IPsec:

- A one-way Telnet using IPsec
- A VPN PPTP tunnel, employing manual key
- A fully meshed VPN topology between three routers, employing manual key
- A fully meshed VPN topology between three routers, employing dynamic key
- A hub and spoke VPN topology between three routers, employing dynamic key

### Creating a Manual Security Policy in Transport Mode

To create a security policy for Telnet traffic using the default encryption algorithm DesCbc between router 1 with IP address 170.0.0.1 to router 2 with IP address 180.0.0.1 (see Figure 146), follow these steps:

**Figure 146** One-way Telnet Using IPsec

- 1 On router 1, enter:

```
ADD !1 -IPSEC manualPOLICY esp_pol EspXport tcp(Telnet,*)(*, Telnet)
170.0.0.1 180.0.0.1
```

- 2 On router 2, enter:

```
ADD !1 -IPSEC manualPOLICY esp_pol EspXport tcp(Telnet,*)(*, Telnet)
180.0.0.1 170.0.0.1
```

To configure a security policy for Telnet traffic using the 3DES encryption algorithm and MD5 authentication from router 1 with IP address 170.0.0.1 to router 2 with IP address 180.0.0.1, follow these steps:



*The following configuration only supports Telnet from 170.0.0.1 to 180.0.0.1 and not in the reverse.*

- 1 On router, 1 enter:

```
ADD !1 -IPSEC manualPOLICY EspAh_pol EspAuthXport tcp(*, Telnet)(Telnet,*)
170.0.0.1 180.0.0.1 3DES MD5
```

- 2 On router 2, enter:

```
ADD !1 -IPSEC manualPOLICY EspAh_pol EspAuthXport tcp(*, Telnet)(Telnet,*)
180.0.0.1 170.0.0.1 3DES MD5
```

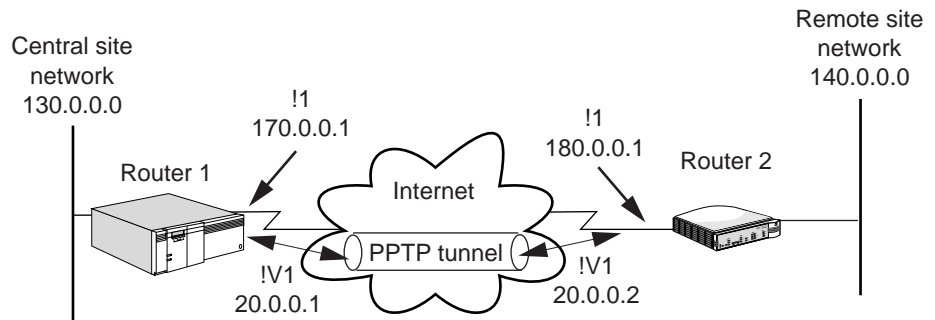
### Manual Key: Setting up a VPN PPTP Tunnel

The procedure that follows shows how to set up a dial VPN PPTP tunnel between router 1 (170.0.0.1) and router 2 (180.0.0.1) with an IPsec policy providing data confidentiality and data integrity, using:

- A PPTP tunnel
- IPsec transport mode and ESP and AH
- Manual policy
- Static routing
- IPsec for all TCP and GRE encapsulated packets



Figure 147 VPN PPTP Tunnel



### Router 1

On router 1, set up the tunnel from 170.0.0.1 to 180.0.0.1 by following these steps:

- 1 Set the system name to "router1" by entering:  
`SETDefault scid = "router1"`
- 2 Create a virtual port to accept connection requests from only router 2 by entering:  
`ADD !v1 -Port VirtualPort scid "router2"`
- 3 Assign an IP address to the tunnel virtual port by entering:  
`SETDefault !v1 -IP NETaddr =20.0.0.1 255.255.0.0`
- 4 Create a route between the two tunnel endpoints by entering:  
`ADD -IP ROute 180.0.0.1 !1 1`
- 5 Create a static router to route traffic over a PPTP tunnel by entering the following or turn on routing protocols on the corresponding virtual port:  
`ADD -IP ROute 140.0.0.0 255.255.0.0 !v1 1`
- 6 Assign peer's dial number to PPTP tunnel dial number list by entering:  
`ADD !v1 -Port DialNoList"@170.0.0.1" Type=pptp`
- 7 Optionally, set the dial idle time-out to zero to keep the tunnel from timing out by entering:  
`SETDefault !v1 -Port DialIdleTime = 0`
- 8 Enable Layer 2 tunnelling by entering:  
`SETDefault -L2Tunnel CONTROL=Enable`
- 9 Erase IP routing by entering:  
`SETDefault -IP CONTROL=ROute`
- 10 Configure an IPsec policy/security association by entering:



*The IPsec policy is a transport mode policy on the physical port. It is not configured on the virtual port for PPTP/L2TP.*

```
ADD !1 -IPSEC manualPOLicy pptp_ahesp EspAhXport tcp,gre 170.0.0.1
180.0.0.1
ADD -IPSEC KeySet pptp_key EncryptKey "Hello572" AuthKey "world329"
```

```
SETDefault !1 -IPSEC ManualKeyInfo=pptp_ahesp pptp_key SpiEsp 500 501
SpiAh 600 601
SETDefault !1 -IPSEC CONTROL=Enable
```

## Router 2

On router 2, set up the PPTP tunnel from 170.0.0.1 to 180.0.0.1 by following these steps:

- 1 Set the system name of router 2 to "router2" by entering:
 

```
SETDefault scid="router2"
```
- 2 Create a virtual port that will accept connection requests from only router1 by entering:
 

```
ADD !v1 -Port VirtualPort scid"router1"
```
- 3 Assign an IP address to the tunnel virtual port by entering:
 

```
SETDefault !v1 -IP NETaddr=20.0.0.2 255.255.0.0
```
- 4 Create a route between two tunnel endpoints by entering:
 

```
ADD -IP ROute 170.0.0.1 !1 1
```
- 5 Add a static route to route traffic over a PPTP tunnel by entering the following or turn on routing protocols on the corresponding virtual port:
 

```
ADD -IP ROute 130.0.0.0 255.255.0.0 !v1 1
```
- 6 Assign the peer dial number to the PPTP tunnel dial number list by entering:
 

```
ADD !v1 -Port DialNoList "@170.0.0.1" Type=pptp
```
- 7 Optionally set dial idle time-out to zero to keep tunnel from timing out by entering:
 

```
SETDefault !v1 -Port DialIdleTime=0
```
- 8 Enable Layer 2 tunnelling (PPTP) by entering:
 

```
SETDefault -L2Tunnel CONTROL=Enable
```
- 9 Erase IP routing by entering:
 

```
SETDefault -IP CONTROL=ROute
```
- 10 Configure an IPsec policy/security association by entering:
 

```
ADD !1 -IPSEC manualPOLicy pptp_ahesp EspAhXport tcp,gre 180.0.0.1
170.0.0.1
ADD -IPSEC KeySet pptp_key EncryptKey "hello572" AuthKey "world329"
SETDefault !1 -IPSEC ManualKeyInfo=pptp_ahesp pptp_key SpiEsp 501 500
SpiAh 601 600
SETDefault !1 -IPSEC CONTROL=Enable
```

## Establishing the Dialup Tunnel

After all the configuration is completed at both ends of the connection, you can dial the PPTP tunnel from either end by entering:

```
Dial !v1
```

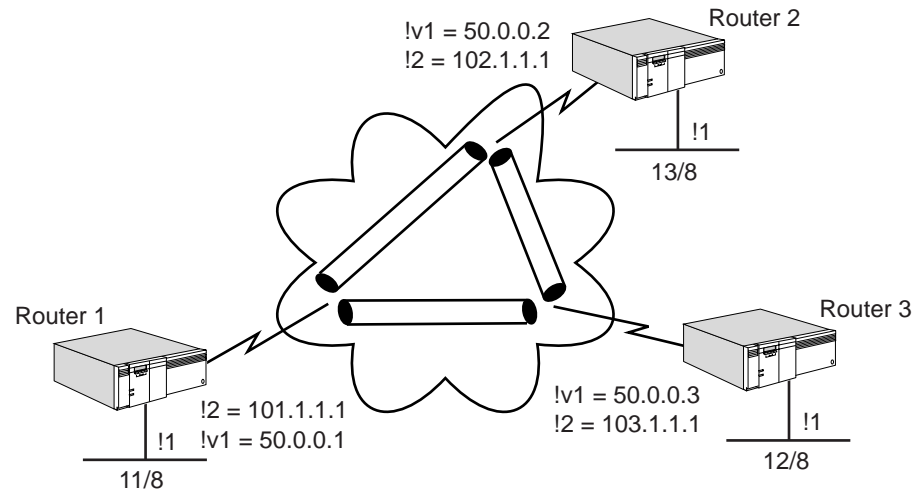
### Manual Key: Creating a Fully Meshed Topology Between Three Routers

This example illustrates a fully meshed topology between three routers, using:

- IPsec tunnel mode for the tunnels.

- ESP for encryption (RC5) and authentication (MD5).
- IPsec manual keys.
- RIP as the routing protocols over the tunnels.

**Figure 148** Manual Key: Fully Meshed Topology Between Three Routers



### Router 1

To configure the router 1 depicted in Figure 148, follow these steps:

- 1 Add an IPIP point-to-multipoint tunnel virtual port by entering:

```
1 ADD !v1 -Port VirtualPort IPIP P2MP
```



*The source IP address of the tunnel is not specified so the outgoing interface IP is used (101.1.1.1).*

- 2 Assign an IP address to the local LAN interface by entering:

```
SETDefault !1 -IP NETaddress = 11.0.0.1
```

- 3 Assign an IP network address to the Internet interface by entering:

```
SETDefault !2 -IP NETaddress = 101.1.1.1
```

- 4 Assign an IP network address to the IPIP P2MP tunnel interface by entering:

```
SETDefault !v1 -IP NETaddress = 50.0.0.1
```

- 5 Specify the mappings of the peer tunnel IP address to the peer Internet IP address, using the following interface IP addresses:

- a For router 2, enter:

```
ADD -IP ADDRESS 50.0.0.2 ipip 102.1.1.1
```

- b For router 3, enter:

```
ADD -IP ADDRESS 50.0.0.3 ipip 103.1.1.1
```

- 6 Add a default route to the Internet (assuming !2 is a PPP port) by entering:

```
ADD -IP ROute 0.0.0.0 !2 1
```

- 7 Enable IP routing by entering:

```
SETDefault -IP CoNTrol = ROute
```

- 8 Configure the IP security information.

- a Configure an IPsec manual policy on the tunnel port (see How IPsec Works earlier in this chapter), by entering:

```
ADD !v1 -IPSEC manualPOLicy pol_eat eat default rc5 md5
```



*This policy uses RC5 for encryption and MD5 for authentication. All traffic over the virtual port (default) will match this policy.*

- b Configure the encryption and authentication keys (see “Configuring Dynamic-Key Security Policies” earlier in this chapter) by entering:

```
ADD -IPSEC KeySet ks_ea ek "ek12345678" ak "ak12345678"
```

- c Bind the keys to the policies and configure the SPIs (see Creating Manual Policies earlier in this chapter) by entering:

```
SETDefault !v1 -IPSEC ManualKeyInfo pol_eat 102.1.1.1 ks_ea se 500 501
SETDefault !v1 -IPSEC ManualKeyInfo pol_eat 103.1.1.1 ks_ea se 500 501
```



*Since ESP is not used for authentication, a Spi\_ah value is not needed.*

- 9 Enable IPsec control on the tunnel port by entering:

```
SETDefault !v1 -IPSEC CONTROL = e
```

- 10 Check the configuration, by entering:

```
SHow -IPSEC CONFiguration
```

- 11 Enable RIP Talk and Listen on the tunnel port by entering:

```
SETDefault !v1 -RIP CONTROL= (ta, li)
```

## Router 2

To configure the router 2 depicted in Figure 148, perform the steps in “Router 1” entering the following information:

```
ADD !v1 -Port VirtualPort IPIP P2MP
SETDefault !1 -IP NETAddress = 12.0.0.1
SETDefault !2 -IP NETAddress = 102.1.1.1
SETDefault !v1 -IP NETAddress = 50.0.0.2
ADD -IP ADDRESS 50.0.0.1 IPIP 101.1.1.1
ADD -IP ADDRESS 50.0.0.3 IPIP 103.1.1.1
ADD -IP ROUTe 0.0.0.0 !2 1
SETDefault -IP CONTROL = ROUTe
ADD !v1 -IPSEC manualPOLicy pol_eat eat default rc5 md5
ADD -IPSEC KeySet ks_ea ek "ek12345678" ak "ak12345678"
SETDefault !v1 -IPSEC ManualKeyInfo = pol_eat 101.1.1.1 ks_ea se 501 500
SETDefault !v1 -IPSEC ManualKeyInfo = pol_eat 103.1.1.1 ks_ea se 600 601
SETDefault !v1 -IPSEC CONTROL = e
SETDefault !v1 -RIP CONTROL = (ta, li)
```

## Router 3

To configure the router 3 depicted in Figure 148, perform the steps in “Router 1” entering the following information:

```
ADD !v1 -Port VirtualPort IPIP P2MP
SETDefault !1 -IP NETAddress = 13.0.0.1
SETDefault !2 -IP NETAddress = 103.1.1.1
SETDefault !v1 -IP NETAddress = 50.0.0.3
ADD -IP ADDRESS 50.0.0.1 IPIP 101.1.1.1
ADD -IP ADDRESS 50.0.0.2 IPIP 102.1.1.1
ADD -IP ROUTe 0.0.0.0 !2 1
SETDefault -IP CONTROL = ROUTe
```

```

SETDefault !v1 -IPSEC manualPOLicy pol_eat eat default rc5 md5
ADD -IPSEC KeySet ks_ea ek "ek12345678" ak "ak12345678"
SETDefault !v1 -IPSEC ManualKeyInfo= pol_eat 101.1.1.1 ks_ea se 501 500
SETDefault !v1 -IPSEC ManualKeyInfo= pol_eat 102.1.1.1 ks_ea se 601 600
SETDefault !v1 -IPSEC CONTROL= e
SETDefault !v1 -RIP CONTROL= (ta, li)

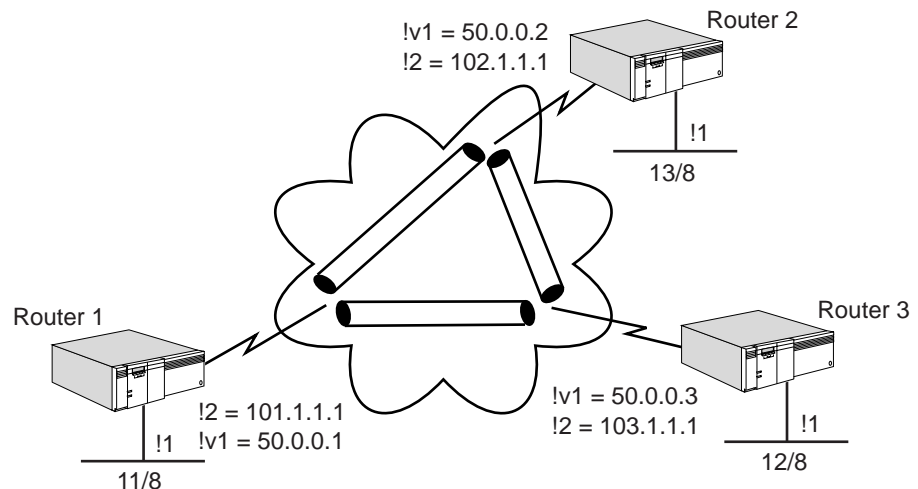
```

### Dynamic Key: Creating a Fully Meshed Topology Between Three Routers

This example illustrates a fully meshed topology between three routers, using:

- IPsec Tunnel mode for the tunnels.
- Dynamic Keys using IKE.
- Preshared keys, DES, MD5 for Phase 1 IKE Profile.
- ESP for encryption (RC5) and authentication (MD5) for Phase 2 TransformList.
- RIP as the routing protocols over the tunnels.

**Figure 149** Dynamic Key: Fully Meshed Topology Between Three Routers



#### Router 1

To configure the router 1 depicted in Figure 149, follow these steps:

- 1 Add an IPIP point-to-multipoint tunnel virtual port by entering:

```
ADD !v1 -Port Virtual Port IPIP P2MP
```

- 2 Assign an IP address to the local LAN interface by entering:

```
SETDefault !1 -IP NETaddress = 11.0.0.1
```

- 3 Assign an IP network address to the Internet interface by entering:

```
SETDefault !2 -IP NETaddress = 101.1.1.1
```

- 4 Assign an IP network address to the IPIP P2MP tunnel interface by entering:

```
SETDefault !v1 -IP NETaddress = 50.0.0.1
```

- 5 Specify the mappings of the peer Tunnel IP address to the peer Internet interface IP addresses using the following interface IP addresses:

- a For router 2, enter:

```
ADD -IP ADDRESS 50.0.0.2 ipip 102.1.1.1
```

- b** For router 3, enter:
- ```
ADD -IP ADDRESS 50.0.0.3 ipip 103.1.1.1
```
- 6** Add a default route to the Internet (assuming !2 is a PPP port) by entering:
- ```
ADD -IP ROUTE 0.0.0.0 !2 1
```
- 7** Enable IP routing by entering:
- ```
SETDefault -IP CONTROL = ROUTE
```
- 8** Configure the IP security information.
- a** Add a selector list to choose which Traffic the policies will apply to. In this case, all traffic over the tunnel is to be encrypted, so the values of 0.0.0.0/0 are used. Enter:
- ```
ADD -IPSEC SelectorList s110 10 include any 0.0.0.0/0 0.0.0.0/0
```
- b** Add a transform list that specifies the Phase 2 SA. (This is the description of the security for the actual data packets over the tunnel.) Enter:
- ```
ADD TransformList t110 10 ESP-RC5 ESP-MD5
```
- c** Define a common preshared key shared by all routers that need to communicate with each other. In this case, mask 0.0.0.0/0 is used to select all routers. Enter:
- ```
ADD PreSharedKey 0.0.0.0/0 "secretkey"
```
- d** Define an IKE profile that describes the Phase 1 SA. This is used by IKE to secure its own negotiation, and is not used to secure the data traffic. Enter:
- ```
ADD IKEProfile 10 PreSharedKey des md5
```
- e** Bind all the information together using a DynamicPOLicy by entering:
- ```
ADD !v1 DynamicPOLicy pol_ea10 10 Tunnel s110 t110
```
- f** Enable IPsec Control on the tunnel port by entering:
- ```
SETDefault !v1 -IPSEC CONTROL = e
```
- g** Check the IPsec configuration by entering:
- ```
SHOW -IPSEC CONFIGURATION
```
- 9** Enable RIP Talk and Listen on the tunnel port by entering:
- ```
SETDefault !v1 -RIP CONTROL= (ta, li)
```

Router 2

To configure the router 2 depicted in Figure 149, follow steps 1 through 10 in "Router 1" entering the following information:

```
ADD !v1 -PORT VirtualPort IPIP P2MP
SETDefault !1 -IP NETaddress = 12.0.0.1
SETDefault !2 -IP NETaddress = 102.1.1.1
SETDefault !v1 -IP NETaddress = 50.0.0.2
ADD -IP ADDRESS 50.0.0.1 IPIP 101.1.1.1
ADD -IP ADDRESS 50.0.0.3 IPIP 103.1.1.1
ADD -IP ROUTE 0.0.0.0 !2 1
SETDefault -IP CONTROL = ROUTE
```

- 10** Configure the IP Security information.

- a Add a SelectorList to choose which Traffic the policies will apply to. In this case all traffic over the Tunnel is to be encrypted, so the values 0.0.0.0/ are used. Enter:

```
ADD -IPSEC SelectorList s110 10 include any 0.0.0.0/0 0.0.0.0/0
```

- b Add a transform list that specifies the Phase 2 SA. This is the description of the security for the actual data packets over the tunnel. Enter:

```
ADD TransformList t110 10 ESP-RC5 ESP-MD5
```

- c Define a common preshared key shared by all routers that need to communicate. In this case, the mask 0.0.0.0/0 is used to select all routers. Enter:

```
ADD PreSharedKey 0.0.0.0/0 "secretkey"
```

- d Define an IKE profile that describes the Phase 1 SA. This is used by IKE to secure its own negotiation and is not used to secure the data traffic. Enter:

```
ADD IKEProfile 10 PreSharedKey des md5
```

- e Bind all the information together using a DynamicPOLicy by entering:

```
ADD !v1 DynamicPOLicy pol_ea10 10 Tunnel s110 t110
```

- f Enable IPsec Control on the Tunnel port by entering:

```
SETDefault !v1 -IPSEC CONTROL = e
```

- g Check the IPsec configuration by entering:

```
SHOW -IPSEC CONFIGuration
SETDefault !v1 -RIP CONTROL= (ta, li)
```

Router 3

To configure the router 3 depicted in Figure 149, follow steps 1 through 10 in "Router 1" entering the following information:

```
ADD !v1 -Port VirtualPort IPIP P2MP
SETDefault !1 -IP NETaddress = 13.0.0.1
SETDefault !2 -IP NETaddress = 103.1.1.1
SETDefault !v1 -IP NETaddress = 50.0.0.3
ADD -IP ADDRESS 50.0.0.1 IPIP 101.1.1.1
ADD -IP ADDRESS 50.0.0.2 IPIP 102.1.1.1
ADD -IP ROUTe 0.0.0.0 !2 1
SETDefault -IP CONTROL = ROUTe
```

- 11 Configure the IP security information.

- a Add a SelectorList to choose which Traffic the policies will apply to. In this case all traffic over the Tunnel is to be encrypted, so the values 0.0.0.0/ are used. Enter:

```
ADD -IPSEC SelectorList s110 10 include any 0.0.0.0/0 0.0.0.0/0
```

- b Add a transform list that specifies the Phase 2 SA. This is the description of the security for the actual data packets over the tunnel. Enter:

```
ADD TransformList t110 10 ESP-RC5 ESP-MD5
```

- c Define a common preshared key shared by all routers that need to communicate. In this case, the mask 0.0.0.0/0 is used to select all routers. Enter:

```
ADD PreSharedKey 0.0.0.0/0 "secretkey"
```

- d Define an IKE profile that describes the Phase 1 SA. This is used by IKE to secure its own negotiation and is not used to secure the data traffic. Enter:

```
ADD IKEProfile 10 PreSharedKey des md5
```

- e Bind all the information together using a DynamicPOLicy by entering:

```
ADD !v1 DynamicPOLicy pol_ea10 10 Tunnel s110 t110
```

- f Enable IPsec Control on the Tunnel port by entering:

```
SETDefault !v1 -IPSEC CONTROL = e
```

- g Check the IPsec configuration by entering:

```
SHow -IPSEC conf
```

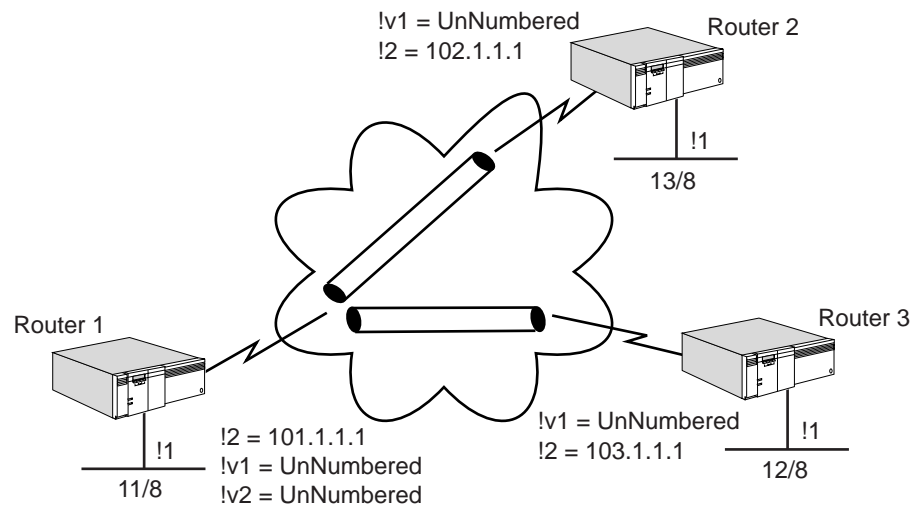
```
SETDefault !v1 -RIP CONTROL = (ta, li)
```

Dynamic Key: Hub and Spoke Topology Between Three Routers

This example illustrates a hub and spoke topology between three routers, using:

- L2TP or PPTP for tunnels.
- IPsec Transport mode.
- Dynamic Keys using IKE.
- Phase 1 IKE Profile using preshared keys, DES, MD5.
- Phase 2 TransformList using ESP for encryption (RC5).
- OSPF as the routing protocols over the tunnels.

Figure 150 Dynamic Key: Hub and Spoke Topology Between Three Routers



Router 1, Router 2, and Router 3

To configure the routers depicted in Figure 150, follow these steps:



All three routers should be configured identically, except where noted in the following procedure.

- 1 Configure PPTP or L2TP tunnels for the topology depicted in Figure 150, using the procedure outlined in the Configuring L2Tunnel Connections chapter.
- 2 Configure the routing policies by entering:

- a Add a default route to the Internet (assuming !2 is a PPP port) by entering:

```
ADD -IP ROute 0.0.0.0 !2 1
```
- b Enable IP routing by entering:

```
SETDefault -IP CONTrol = ROute
```
- 3 Configure the IPSec policy by entering:
 - a Add a SelectorList to choose that traffic the policies will apply to. In this case all traffic over the Internet port is to be encrypted, so the values 0.0.0.0/ are used. Enter:

```
ADD -IPSEC SelectorLIst s110 10 include any 0.0.0.0/0 0.0.0.0/0
```
 - b Add a transform list that specifies the Phase 2 SA. This is the description of the security for the actual data packets over the tunnel. Enter:

```
ADD TransformLIst t110 10 ESP-RC5 ESP-MD5
```
 - c Define a common preshared key shared by all routers that need to communicate. In this case, the mask 0.0.0.0/0 is used to select all routers. Enter:

```
ADD PreSharedKey 0.0.0.0/0 "secretkey1234567"
```
 - d Define an IKE profile that describes the Phase 1 SA. This is used by IKE to secure its own negotiation and is not used to secure the data traffic. Enter:

```
ADD IKEProfile 10 PreSharedKey des md5
```
 - e Bind all the information together using a DynamicPOLicy by entering:

```
ADD !2 DynamicPOLicy pol_ea10 10 Xport s110 t110
```



For PPTP/L2TP using IPSec transport mode, this needs to be configured on the actual physical port, not the virtual port.

- f Enable IPSec Control on the IPSec port by entering:

```
SETDefault !2 -IPSEC CONTrol= e
```
- g Check the IPSec configuration by entering:

```
SHow -IPSEC CONFiguration
```
- 4 Enable OSPF on the virtual ports by entering:
 - a For router 1, enter:

```
SETDefault !v1 -Ospf CONTrol = e  
SETDefault !v2 -Ospf CONTrol = e
```
 - b For router 2, enter:

```
SETDefault !v1 -Ospf CONTrol = e
```
 - c For router 3, enter:

```
SETDefault !v1 -Ospf CONTrol = e
```

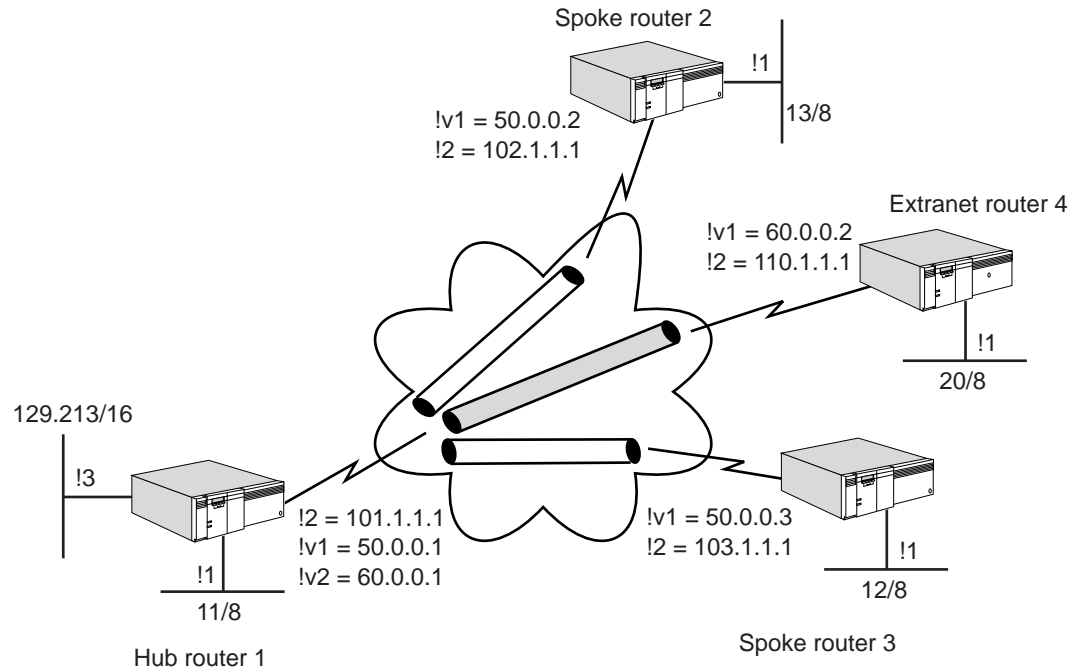


This assumes that port !2 is not running OSPF and direct policy is not configured.

Dynamic Key: Hub and Spoke Topology Between Three Routers (Intranet/Extranet)

This example illustrates a hub and spoke topology between three routers that constitute an intranet, as all routers belong to the same organization. Additionally, creation of a tunnel from the hub router to an extranet router is illustrated. The extranet router belongs to a different organization. See Figure 151.

Figure 151 Dynamic Key: Hub and Spoke Topology Between (Intranet/Extranet)



The following configuration properties are demonstrated in this example:

- The hub router has an IPIP P2MP tunnel connected to its Intranet spoke routers.
- The intranet spoke routers have a P2P tunnel connected to the hub.
- The hub router has a P2P tunnel over a separate virtual port to the extranet router. (It is best to use a separate virtual port for the extranet router as this makes configuring policies simpler, and there is less chance of creating a security hole.)
- IPsec Tunnel mode is used with IKE.
- Hub to intranet routers: IPsec on all OSPF traffic.
- Hub to extranet: IPsec on all data and RIP traffic.
- Hub to intranet router R2: ESP-3DES ESP-MD5.
- Hub to intranet router R3: ESP-DES ESP-MD5.
- An IPsec GlobalLifeTime of 30-minutes is used.
- With the intranet routers, the IKEProfiles have a lifetime of 6-hours and Group1 PFS.
- Hub to extranet router R4: ESP-RC5 and ESP-SHA.
- With the extranet router, the IKEProfile uses 3DES, MD5, Group2 PFS.
- OSPF is used for routing over the intranet.
- RIPv2 is used for routing to the extranet router.
- There should be complete connectivity in the intranet.
- The extranet router should only see network 129.213/16.

Spoke Router 1

To configure the spoke router 1 depicted in Figure 151, follow these steps:

- 1 Configure the system prompt by entering:


```
SETDefault -SYS NMPrompt = "HubRtr1 # "
```
 - 2 Assign an IP address to the local LAN interface by entering:


```
SETDefault !1 -IP NETaddr = 11.0.0.1
```
 - 3 Assign an IP address to the Internet interface by entering:


```
SETDefault !2 -IP NETaddr = 101.1.1.1
```
 - 4 Assign an IP address to interface that is exposed to extranet by entering:


```
SETDefault !3 -IP NETaddr = 129.213.1.1
```
 - 5 Add an IPIP point-to-multipoint virtual port for the intranet by entering:



```
ADD !v1 -PORT VirtualPort IPIP P2MP
SETDefault !v1 -IP NETaddr = 50.0.0.1
```
 - 6 Add an IPIP point-to-Point virtual port for the extranet router by entering:


```
ADD !v2 -PORT virtualPort ipip 110.1.1.1
SETDefault !v2 -IP NETaddr = 60.0.0.1
```
 - 7 Specify the mappings of the peer tunnel IP address to the peer internet interface IP addresses for the intranet routers by entering:


```
ADD -IP ADDRESS 50.0.0.2 IPIP 102.1.1.1
ADD -IP ADDRESS 50.0.0.3 IPIP 103.1.1.1
```
 - 8 Add a default route to the internet (assuming !2 is a PPP port) by entering:


```
ADD -IP ROUTe 0.0.0.0 !2 1
```
 - 9 Enable IP routing by entering:


```
SETDefault -IP CONTROL = ROUTe
```
 - 10 Enable OSPF for the intranet by entering:


```
SETDefault !v1 -OSPF CONTROL = e
```
-  *There is no need to configure OSPF neighbors. They are automatically picked up from the ADD -IP ADDRESS configuration.*
- 11 Enable RIPv2 on the extranet interface by entering:


```
SETDefault !v2 -RIP CONTROL= (talk, listen)
SETDefault !v2 -RIP v2cm = ripv2
```
 - 12 Configure the IP security information.
 - a Set GlobalLifeTime so that IPsec SA's are re-negotiated every 30-minutes by entering:


```
SETDefault -IPSEC GlobalLifeTime = 30m
```
 - b Add a SelectorList to choose all traffic by entering:


```
ADD -IPSEC SelectorList slany 100 include any 0.0.0.0 0.0.0.0
```
 - c Add a TransformList that specifies all the transforms the hub offers to the intranet routers by entering:


```
ADD -IPSEC TransformList tlintra 10 esp-3des esp-md5
ADD -IPSEC TransformList tlintra 20 esp-des esp-md5
```

- d Add a TransformList that specifies the transforms the hub offers to the extranet routers by entering:

```
ADD -IPSEC TransformList tlextra 100 esp-RC5 esp-md5
```

- e Add a preshared key for the intranet routers by entering:

```
ADD -IPSEC PreSharedKey 110.1.1.1 "secretExtranet"
```

- f Add a preshared key for the extranet router by entering:

```
ADD -IPSEC PreSharedKey 0.0.0.0 "secretIntranet"
```

- g Define an IKEProfile for the extranet router by entering:

```
ADD IKEProfile 10 psk 3des md5 g2
```

- h Define an IKEProfile for the intranet routers by entering:

```
ADD IKEProfile 20 psk des md5 g1 6h
```

- i Define a DynamicPOLicy for the intranet routers by entering:

```
ADD !v1 -IPSEC DynamicPOLicy dpintra 100 t slany tlintra
```

- j Define a DynamicPOLicy for the extranet router by entering:

```
ADD !v2 -IPSEC DynamicPOLicy dpextranet 500 t slany tlextra
```

- k Enable IPsec by entering:

```
SETDefault !v1 -IPSEC CONTROL = e
```

```
SETDefault !v2 -IPSEC CONTROL = e
```

Spoke Router 2 (Intranet)

To configure the spoke router 2 depicted in Figure 151, follow these steps:

- 1 Configure the system prompt by entering:

```
SETDefault -SYS NMPrompt = "SpkRtr2 # "
```

- 2 Assign an IP address to the local LAN interface by entering:

```
SETDefault !1 -IP NETaddr = 13.0.0.1
```

- 3 Assign an IP address to the internet interface by entering:

```
SETDefault !2 -IP NETaddr = 102.1.1.1
```

- 4 Add an IPIP point-to-point virtual port by entering:

```
ADD !v1 -port virtualPort ipip 101.1.1.1
```

```
SETDefault !v1 -IP NETaddr = 50.0.0.2
```

- 5 Add a default route to the internet (assuming !2 is a PPP port) by entering:

```
ADD -IP ROute 0.0.0.0 !2 1
```

- 6 Enable IP routing by entering:

```
SETDefault -IP CONTROL = ROute
```

- 7 Enable OSPF for the intranet by entering:

```
SETDefault !v1 -OSPF CONTROL = e
```

- 8 Configure the IP security information.

- a Set GlobalLifeTime so that IPsec SA's are re-negotiated every 30-minutes by entering:

```
SETDefault -IPSEC GlobalLifeTime = 30m
```

- b Add a SelectorList to choose all traffic by entering:
`ADD -IPSEC SelectorList slany 100 include any 0.0.0.0 0.0.0.0`
- c Add a TransformList that specifies all the Transforms by entering:
`ADD -IPSEC TransformList tlintra 10 esp-3des esp-md5`
- d Add a preshared key for the intranet routers
`ADD -IPSEC PreSharedKey 0.0.0.0 "secretIntranet"`
- e Define an IKEProfile for the intranet routers by entering:
`ADD IKEProfile 20 psk des md5 g1 6h`
- f Define DynamicPOLicy for the intranet routers by entering:
`ADD !v1 -IPSEC DynamicPOLicy dpintra 100 t slintra tlintra`
- g Enable IPsec by entering:
`SETDefault !v1 -IPSEC CONTROL = e`

Spoke Router 3

- 1 Configure the system prompt by entering:
`SETDefault -SYS NMPrompt = "SpkRtr3 # "`
- 2 Assign an IP address to the local LAN interface by entering:
`SETDefault !1 -IP NETaddr = 12.0.0.1`
- 3 Assign an IP address to the internet interface by entering:
`SETDefault !2 -IP NETaddr = 103.1.1.1`
- 4 Add an IPIP point-to-point virtual port by entering:
`ADD !v21-port virtualPort ipip 101.1.1.1`
`SETDefault !v1 -IP NETaddr = 50.0.0.3`
- 5 Add a default route to the internet (assuming !2 is a PPP port) by entering:
`ADD -IP ROUTe 0.0.0.0 !2 1`
- 6 Enable IP routing by entering:
`SETDefault -IP CONTROL = ROUTe`
- 7 Enable OSPF for the intranet by entering:
`SETDefault !v1 -OSPF CONTROL = e`
- 8 Configure the IP security information.
 - a Set GlobalLifeTime so that IPsec SA's are re-negotiated every 30-minutes by entering:
`SETDefault -IPSEC GlobalLifeTime = 30m`
 - b Add a SelectorList to choose all traffic by entering:
`ADD -IPSEC SelectorList slany 100 include any 0.0.0.0 0.0.0.0`
 - c Add a TransformList that specifies all the transforms by entering:
`ADD -IPSEC TransformList tlintra 20 esp-des esp-md5`
 - d Add a preshared key for the intranet routers by entering:
`ADD -IPSEC PreSharedKey 0.0.0.0 "secretIntranet"`
 - e Define an IKEProfile for the intranet routers by entering:

```
ADD IKEProfile 20 psk des md5 g1 6h
```

- f Define DynamicPOLicy for the Intranet routers by entering:

```
ADD !v1 -IPSEC DynamicPOLicy dpintra 100 t slintra tlintra
```

- g Enable IPsec by entering:

```
SETDefault !v1 -IPSEC CONTROL = e
```

Extranet Router 4

- 1 Configure the system prompt by entering:

```
SETDefault -SYS NMPrompt = "ExtraRtr4 # "
```

- 2 Assign an IP address to the local LAN interface by entering:

```
SETDefault !1 -IP NETAddr = 20.0.0.1
```

- 3 Assign an IP address to the internet interface by entering:

```
SETDefault !2 -IP NETAddr = 110.1.1.1
```

- 4 Add an IPIP point-to-point virtual port by entering:

```
ADD !v1 -port virtualPort ipip 101.1.1.1
SETDefault !v1 -IP NETAddr = 60.0.0.2
```

- 5 Add a default route to the internet (assuming !2 is a PPP port) by entering:

```
ADD -ip ro 0.0.0.0 !2 1
```

- 6 Enable RIPv1 on the extranet Interface by entering:

```
SETDefault !v1 -RIP CONTROL= (talk, listen)
SETDefault !v1 -RIP v2cm = ripv2
```

- 7 Make sure that only 20.0.0.0 is advertised via RIP by entering:

```
Add !v2 -rip AdvertisePol 20.0.0.0
```



It is very important to make sure that the IP network of the Internet interface is NOT advertised over the tunnel. Doing so will cause routing loops and packet loss.

- 8 Enable IP routing

```
SETDefault -IP CONTROL = Route
```

- 9 Configure the IP security information.

- a Set GlobalLifeTime so that IPsec SA's are re-negotiated every 30-minutes by entering:

```
SETDefault -IPSEC GlobalLifeTime = 30m
```

- b Add a SelectorList to choose all traffic by entering:

```
ADD -IPSEC SelectorList slany 100 include any 0.0.0.0 0.0.0.0
```

- c Add a TransformList that specifies the transforms by entering:

```
ADD -IPSEC TransformList tlextra 100 esp-EC8 esp-SHA
```

- d Add a preshared key for the extranet router by entering:

```
ADD -IPSEC PreSharedKey 101.1.1.1 "secretExtranet"
```

- e Define an IKEProfile for the extranet router by entering:

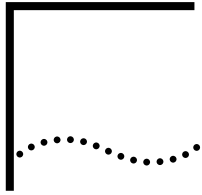
```
ADD IKEProfile 10 psk 3des md5 g2
```

- f Define DynamicPOLicy for the extranet router by entering:

```
ADD !v1 -IPSEC DynamicPOLicy dpextranet 500 t slany tlextra
```

g Enable IPsec by entering:

```
SETDefault !v1 -IPSEC CONTROL = e
```

CONFIGURING APPN INTERMEDIATE SESSION ROUTING

This chapter describes how to configure your 3Com bridge/router to function as a network node in an Advanced Peer-to-Peer Networking (APPN) network.

APPN is an architecture designed to provide peer-to-peer routing services for Systems Network Architecture (SNA) environments. APPN is designed to work with Advanced Program-to-Program Communications (APPC) functions, and APPN uses LU 6.2 sessions to exchange network information between nodes. The 3Com implementation of APPN allows the bridge/router to function as a network node in an APPN network as well as serve as a Dependent LU Requester (DLUR) for relaying sessions with dependent logical units (LUs) on physical unit (PU) type 2.0 and 2.1 nodes.

Two types of APPN routing are available on the NETBuilder II bridge/router:

- Intermediate Session Routing (ISR)
- High Performance Routing (HPR)

This chapter describes how to configure your bridge/router for Intermediate Session Routing only. For information on how to configure your bridge/router as a network node for High Performance Routing, see the APPN High Performance Routing chapter.



CAUTION: *HPR is enabled by default. If you want your existing network to perform ISR only, then you must disable HPR on your APPN ports and adjacent link stations.*



For conceptual information about APPN, see "How APPN ISR Routing Works" later in this chapter.

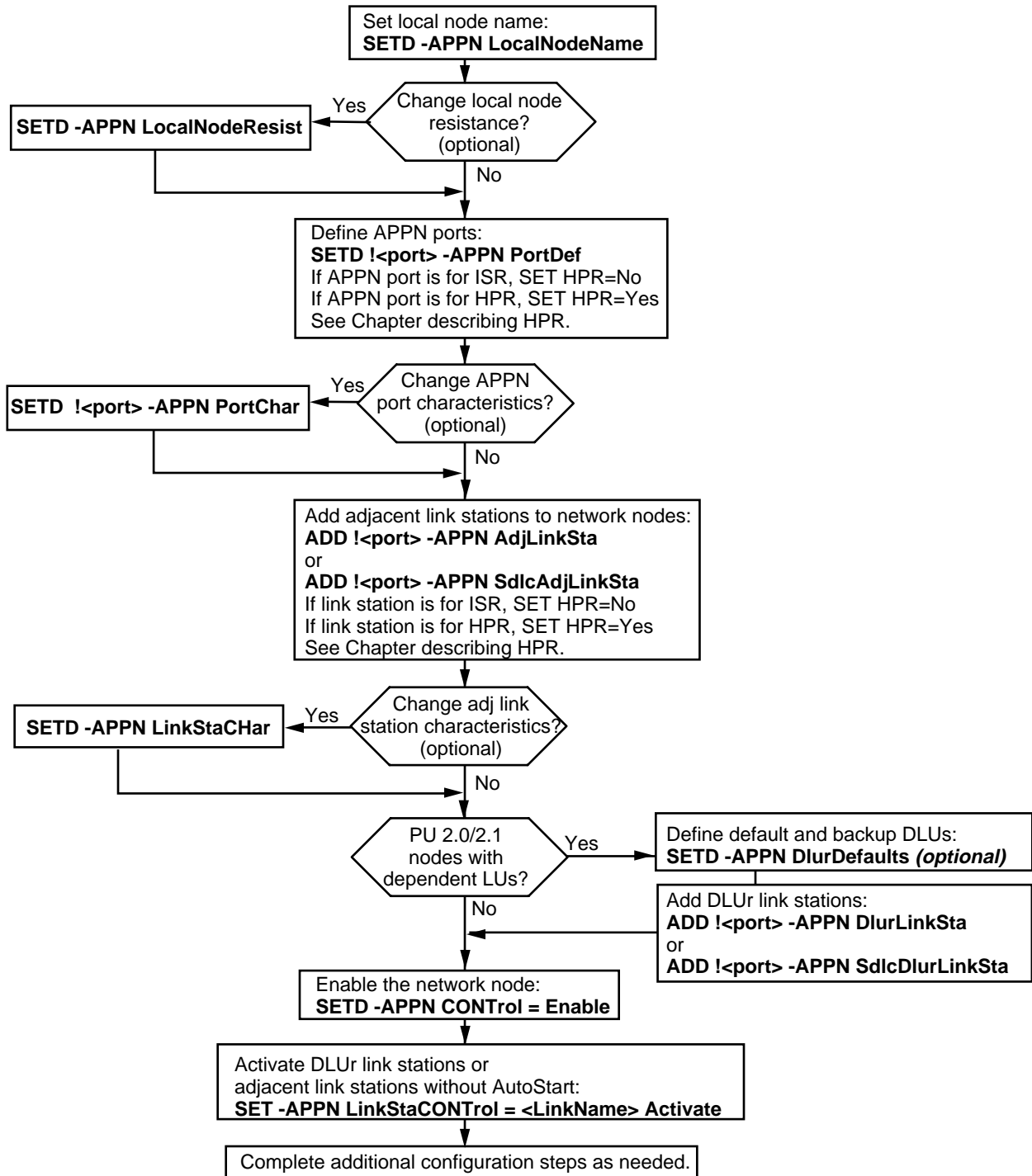
Setting Up a Basic APPN Router

The procedures in this section explain how to configure your bridge/router as a network node and configure node information to initiate APPN routing. The minimum tasks required to configure the APPN network node are separated into the following procedures:

- Setting up the bridge/router as a network node (also referred to as the *local node* when working directly at that node)
- Defining links from your system to adjacent network nodes
- Configuring dependent LU support if you have PU 2.0 nodes and PU 2.1 nodes with dependent LUs in your network
- Enabling the network node

Figure 152 provides a flowchart of the basic steps to configure the bridge/router so that it will operate as an APPN network node.

Figure 152 Basic APPN Configuration Steps



Setting Up Your System as a Network Node

The first task in setting up the APPN environment is to configure the local bridge/router (referred in this section as “local node”) to serve as a network node. The NETBuilder II system can be configured as a network node only; because the bridge/router does not provide any application programs on the SNA network, it cannot act as an end node or LEN end node. Viewed from the SNA network, the bridge/router network node has only one LU for handling CP-CP sessions.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the procedures described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- If necessary, use LAN Address Administration (LAA) to reassign MAC addresses for paths that will be sending and receiving APPN traffic.

You must perform this configuration *before* starting APPN. For more information on configuring LAA, see the Configuring LAN Address Administration chapter.

- If you are planning to support both APPN and DECnet on the same bridge/router, you must configure DECnet *before* configuring APPN. Configuring DECnet can change MAC addresses, which will affect any existing APPN configuration. For more information on configuring DECnet, see the Configuring DECnet Routing chapter.
- If necessary, configure the Logical Link Control type 2 (LLC2) data link interface or the data link switching (DLSw) interface for the ports you will use for APPN traffic. For more information on configuring the LLC2 data link interface, see the Configuring the LLC2 Data Link Interface chapter. For more information on configuring DLSw, see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.



APPN is affected by parameter settings in other services. For more information, see the Configuring the LLC2 Data Link Interface chapter.

- If you will be sending APPN traffic over SDLC lines, configure the bridge/router for SDLC operation first. For more information on SDLC configuration, see the Configuring Synchronous Data Link Control Connectivity chapter.
- If you will be sending APPN traffic over Frame Relay, configure the Frame Relay interface before configuring the APPN network node. For more information on configuring Frame Relay, see the Configuring Wide Area Networking Using Frame Relay chapter.
- If you are not familiar with APPN routing concepts, see “How APPN ISR Routing Works” later in this chapter.
- See the IBM documents describing APPN architecture listed in “IBM APPN References” later in this chapter.

Procedure

To set up the bridge/router as a network node, follow these steps:

- 1 Assign a name to the local node using:

```
SETDefault -APPN LocalNodeName = <netid.cpname> [node_id]
```

This command creates the *fully qualified* control point (CP) name by combining the network ID with the CP name you create to identify the node. The fully qualified CP name identifies the network node throughout the APPN network. (When the CP name is used without the network ID, it is called a *not fully qualified* CP name.) For more information on CP name formats, see “Fully Qualified and Not Fully Qualified CP Name Formats” later in this chapter.

For example, to assign the local node name consisting of the network ID US3COMHQ plus the CP name NB2SF011, enter:

```
SETDefault -APPN LocalNodeName = US3COMHQ.NB2SF011
```



CAUTION: Every fully qualified CP name on the APPN network must be unique.

Optionally, you can add a node ID following the network ID. This node ID is used in XID negotiations. For more information, see the description of the LocalNodeName parameter in the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

- 2 If desired, change the resistance value of the local node using:

```
SETDefault -APPN LocalNodeResist = <node_resistance> (0-255)
```

The resistance value advertises the desirability of routing through the node. Using different values, you can fine-tune your network to set different resistance rates on different nodes so that more traffic is routed over specific nodes.

The value of the LocalNodeResist parameter ranges from 0 to 255. A value of 0 indicates that routing is highly desirable through this node, while a value of 255 indicates routing is not desirable through the node. The default value is 128, or the median. Changing the value is optional.

- 3 Define each local port on the system that will send and receive APPN traffic using:

```
SETDefault !<port> -APPN PortDef = <DLC type>
(LLC2|FR|PPP|DLSW|SDLC|UNdef) <max_btu_size>(99-8192)
[ActLimit=<limit>(1-512)] [TGprof=<name>] [HPR=(Yes|No)]
[ErrorRecovery=(Yes|No)] [DatMode=(Half|Full)] [ROle=(Neg|Pri|Sec)]
```

Use this command to define the type of traffic being sent over the port (DLC type), as well as the maximum basic transmission unit (BTU) size the port will allow. To define the DLC type, enter LLC2 for token ring, Ethernet, FDDI and PPP links. Enter FR for Frame Relay, or DLSw for using Data Link Switching over an IP network. If you specify the DLC type as DLSw, the port number specified must be !0. Do not specify !0 if using a DLC type other than DLSw. Enter SDLC if you will be sending traffic to and from SDLC devices. Enter UNdef to remove a previously-defined port definition DLC type.



If a port has already been defined for a particular DLC type, the port definition must be removed by setting the DLC type to UNdef before it can be changed to another DLC type.

To determine the maximum BTU size to use, first determine the appropriate request/response unit (RU) size, then add an additional nine bytes (three bytes for the request header (RH) plus six bytes for the transmission header (TH)). The RU size plus the additional nine bytes comprise the BTU size. For more information on the

values for the PortDef parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

For more information about setting the maximum BTU size, see “Setting the Maximum BTU Size” later in this chapter.

Optionally, you can set the activation limit (total number of LLC2 sessions for the port), and if desired, a transmission group (TG) profile for the port. For more information on TG profiles you can use, see the description of the AdjLinkSta parameter in the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.



CAUTION: The PortDef parameter has an option to provide support for High Performance Routing. The default value for the HPR option is Yes, meaning that HPR is automatically enabled. If you want the port to perform Intermediate Session Routing (ISR) only, you must disable the HPR option by typing HPR=No as part of the command. If you want the port to perform HPR, do not change the HPR value, but note that the functionality and routing methods of HPR may be different from ISR. If you have links between two network nodes with HPR enabled, this configuration will create an HPR subnet in your ISR network. For more information about HPR, see the APPN High Performance Routing chapter.

For example, to configure port 7 as an APPN ISR port to handle Frame Relay traffic with a maximum BTU size of 1033, an activation limit of 128, and to use the TG profile SER256, enter:

```
SETDefault !7 -APPN PortDef = FR 1033 ActLimit=128 TGprof=SER256 HPR=No
```

If you specify synchronous data link control (SDLC) as your DLC type, you can specify the DatMode value to either half duplex or full duplex, and you can specify whether the SDLC port will be the primary or secondary device in session negotiation, or whether the role will be negotiable. If you set your DLC type to SDLC, when configuring SDLC devices as adjacent link stations or as DLUR link stations you must use the SdlcAdjLinkSta or SdlcDlurLinkSta parameters, respectively.

For example, to configure port 6 as an APPN ISR port to handle SDLC traffic you can set the following attributes: maximum BTU size of 1033, activation limit of 254, TG profile of Ser19.6, and full duplex data transmission mode. To configure these attributes and set the local node as the primary device in session negotiation, enter:

```
SETDefault !6 -APPN PortDef = SDLC 1033 ActLimit=254 TGProf=Ser19.2 HPR=No
  DatMode=Full Role=Pri
```

Repeat this step for each port on the system used to send and receive APPN sessions.

- 4 If desired, define the characteristics of each APPN port configured in the previous step using:

```
SETDefault !<port> -APPN PortChar = [EffectCap=<string>]
  [ConnectCost=<0-255>] [ByteCost=<0-255>] [Security=<string>]
  [PropDelay=<string>] [Usd1=<0-255>] [Usd2=<0-255>] [Usd3=<0-255>]
```

Using this parameter, you can specify optional settings for the port's effective capacity, connection cost, byte cost, propagation delay, and three user-configurable settings. For more information on the PortChar parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

- 5 Repeat this procedure for each bridge/router functioning as a network node in your APPN network.

After you have set up the bridge/router as a network node, you must then define the links to other network nodes in the APPN network. Proceed to the next section.

Defining Links to Other Network Nodes

After you have performed the basic configuration of the local node, the next step is to define the adjacent link stations to other network nodes. An *adjacent link station* is the local information regarding a link to an adjacent node. The adjacent link station is the link definition, or the representation of the link as seen by the network node.

Two network nodes that connect and exchange data are called *partner nodes*. To configure an adjacent network node as a partner node, you must configure an adjacent link station to the other node; in this situation, the other network node does not need to configure an adjacent link station to your local node. Only one of the partner nodes needs to configure the other as an adjacent link station.



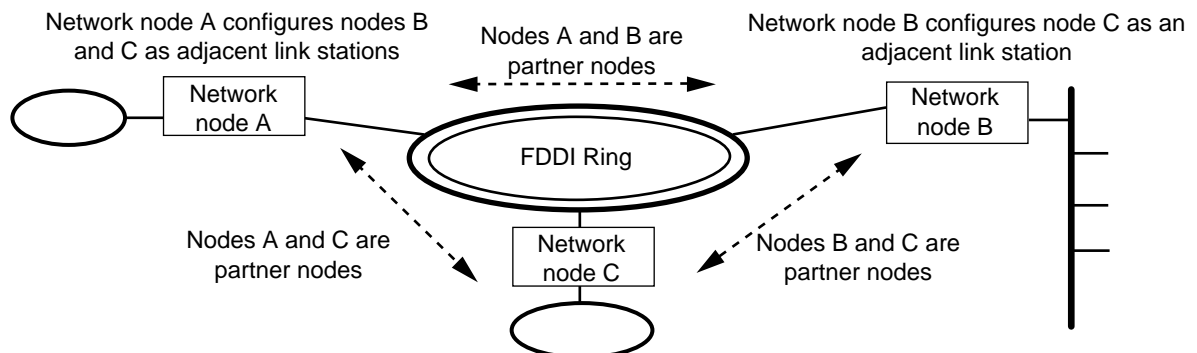
You can add links to other network nodes dynamically after the network node is enabled. For more information on dynamic configuration, see "Dynamic Configuration Options" later in this chapter.

Figure 153 is an example of a network with three different network nodes, each with its own local network, on a larger FDDI ring. In this topology, network nodes A and B are partner nodes to each other, network nodes A and C are partner nodes, and network nodes B and C are partner nodes.

For each of these partner node pairs, only one network node needs to configure its partner as an adjacent link station if both nodes are NETBuilder II bridge/routers. If one of the partner nodes is not a NETBuilder II bridge/router, the links may need to be configured in both directions, depending on the device.

For example, if network node A configures node B as an adjacent link station, then network node B does not also need to configure node A as an adjacent link station. If both partner nodes are 3Com bridge/routers, this situation applies. You can configure links in both directions, but it is not required.

Figure 153 Network Nodes as Adjacent Link Stations (Example)



Because network node C has been configured as an adjacent link station from nodes A and B, node C does not have to configure either A or B as an adjacent link station

Procedure

To define adjacent link stations to partner network nodes, follow these steps:

- 1 If you set the port DLC type (configured with step 3 of the previous procedure) to LLC2, FR, PPP, or DLSw, go to step a. If you set the port DLC type to SDLC in the previous procedure, go to step b.
 - a If you previously set the port DLC type to LLC2, FR, PPP, or DLSW, define the adjacent link station using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn)
    <max_btu_size>(99-8912) [[Cmac|Ncmac] dest media addr] [Sap=<num>]
    [CPName=[netid.]cpname] [Nodeid=<ID>] [LinkName=<name>]
    [TGprof=<name>] [AutoStart=(Yes|No)] [CPSess=(Yes|No)] [HPR=(Yes|No)]
    [ErrorRecovery=(Yes|No)]
```

Make sure you specify the node type as NN. In addition, specify the maximum BTU byte size and the media address of the destination node (or DLCI if running Frame Relay over a virtual port). Optionally, you can set the following for the destination node: the CP name and the node name, the node ID, the link name, the TG profile, whether the link will support AutoStart, and whether control point-to-control point (CP-CP) sessions will be activated with the adjacent node. For more information on the AdjLinkSta parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.



CAUTION: The AdjLinkSta parameter has an option to provide support for High Performance Routing. The default value for the HPR option is Yes, meaning that HPR is automatically enabled. If you want the link station to support Intermediate Session Routing (ISR) only, you must disable the HPR option by typing HPR=No as part of the command, and you also must disable HPR on the port by specifying HPR=No as part of the SETDefault !<port>-APPN PortDef command. If you only disable HPR on the adjacent link station but not the port, then HPR will not be totally disabled for APPN connections. If you want the link station to support HPR, do not change the HPR value, but note that the functionality and routing methods of HPR may be different from ISR. For more information about HPR, see the APPN High Performance Routing chapter.

If you do not define a link name, then the local network node will assign a unique link name to the link. (You will need the link name to complete step 2. If you do not assign a link name, you can obtain the link names assigned by the system using the SHow -APPN LinkStaCONTRol command.)

For example, to add a link to an ISR network node named "FINANCE" to port 3 with a maximum BTU size of 1033 (specifying the appropriate MAC address and SAP) and a fully-qualified CP name "HQ.Finance" (with a link named FINANCE3), profile SER64, and to activate a CP-CP session when the node comes up, enter:

```
ADD !3 -APPN AdjLinkSta NN 1033 N100040C08ACE Sap=08 CPName=HQ.FINANCE
    LinkName=FINANCE3 TGprof=SER64 CPSess=Yes HPR=No
```

For information on how to obtain the MAC address of the node, see the documentation for the end node device or applications.

To obtain the MAC address of another 3Com bridge/router acting as a network node, enter the SHow -SYS Configuration command on the second bridge/router. Enter the MAC address of the port number over which the link is established, making sure to enter the address in the correct format.



If you set the `-SYS MacAddrFmt` parameter to `noncanonical`, then you do not need to precede the MAC address with `N` or `Ncmac`. If you do not change the `-SYS MacAddrFmt` parameter, then the default will be `canonical`, and you will need to precede the MAC address with `N` for `noncanonical` format. If the `-SYS MacAddrFmt` parameter is set to `Default`, then the system will assume that the MAC address is in `noncanonical` format for `token ring` and `FDDI` ports, and `canonical` format for all other port types. For more information on MAC address format options for APPN, see “MAC Address Format Options for APPN” later in this chapter.

- b** If you previously set the port DLC type to `SDLC`, define the `SDLC` adjacent link station using:

```
ADD !<port> -APPN SdlcAdjLinkSta <type>(NN|EN|Learn)
    <max_btu_size>(99-8912) <station_addr>(Hex 1-FE)
    [CPName=<[netid.]cpname>] [Nodeid=<ID>] [LinkName=<name>]
    [TGprof=<name>] [AutoStart=(Yes|No)] [CPSess=(Yes|No)]
    [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)] [SendWindow=<num>]
    [ContactTimer=<num>] [NoRspTimer=<num>]
    [NoRspTimRetry=<num>]
```

Make sure you specify the node type as `NN`. In addition, specify the maximum BTU byte size and the station address of the destination node. Optionally, you can set the CP name of the destination node and the node name, the node ID, the link name, the TG profile, whether the link will support `AutoStart`, and whether CP-CP sessions will be activated with the adjacent node. You can also set the `SDLC` `SendWindow`, `ContactTimer`, `NoRspTimer`, and `NoRspTimRetry` values. You can enter these options in any combination. The default value for `AutoStart` is `yes`, which means when you enable the network node, the link will be activated automatically. For the `SDLC` connection to take place, both `SDLC` partner nodes must be configured as `SDLC` adjacent link stations using the `SdlcAdjLinkSta` parameter.

For more information on the `SdlcAdjLinkSta` parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.



CAUTION: The `SdlcAdjLinkSta` parameter has an option to provide support for `High Performance Routing`. The default value for the `HPR` option is `Yes`, meaning that `HPR` is automatically enabled. If you want the link station to support `Intermediate Session Routing (ISR)` only, you must disable the `HPR` option by typing `HPR=No` as part of the command, and you must also disable `HPR` on the port by specifying `HPR=No` as part of the `SETDefault !<port>-APPN PortDef` command. If you only disable `HPR` on the adjacent link station but not the port, then `HPR` will not be totally disabled for `SDLC` connections. If you want the link station to support `HPR`, do not change the `HPR` value, but note that the functionality and routing methods of `HPR` may be different from `ISR`. Note also that for `HPR` over `SDLC` to work properly, `HPR` must be configured on both partner network nodes. For more information about `HPR`, see the *APPN High Performance Routing* chapter.

If you do not define a link name, then the local network node will assign a unique link name to the link. (You will need the link name to complete step 2. If you do not assign a link name, you can obtain the link names assigned by the system using the `SHoW -APPN LinkStaCONTRol` command.)

For example, to add an `SDLC` link named “`SDLC001`” on port 4 to a network node named “`HQ.FINANCE`” you can set the following attributes: a station address of hex `FE`, maximum BTU size of 1033, TGprofile `SER64`, activation of a CP-CP session when the node comes up, no support for `HPR`, `SendWindow`

size of 4, ContactTimer setting of 2 seconds, NoRspTimer setting of 2000 milliseconds, and a NoRspTimRetry setting of 6. To add this link and configure the attributes, enter:

```
ADD !4 -APPN SdlcAdjLinkSta NN 1033 FE CPName=HQ.FINANCE
      LinkName=SDLC001 TGprof=SER64 CPSess=Yes HPR=No SendWindow=4
      ContactTimer=2 NoRspTimer=2000 NoRspTimRetry=6
```

The ContactTimer, NoRspTimer and NoRspTimRetry values are valid only if the local network node is the primary station on the SDLC link. Also, The SDLC link must be configured before configuring APPN over SDLC. For more information on SDLC, see the Configuring Synchronous Data Link Control Connectivity chapter.



APPN over SDLC connections is supported on all types of HSS 3-Port modules, including V.35, RS-232, and RS-449.

When you configure SDLC adjacent link stations for APPN, if an active link becomes inactive and you change the port definition using the PortDef parameter, the link remains inactive. If you try to reactivate the link using the SET -APPN LinkStaCONTROL command, the link will reactivate within 30 seconds. To activate the link immediately, you must enable the APPN port using the SET -APPN PortControl = Enable command.

- 2 After you have defined the link to the adjacent network node, you define the characteristics of the link using:

```
SETDefault -APPN LinkStaChar = <LinkStation name>
  [EffectCap=<string>] [ConnectCost=<0-255>] [ByteCost=<0-255>]
  [Security=<string>] [PropDelay=<string>] [Usd1=<0-255>]
  [Usd2=<0-255>] [Usd3=<0-255>]
```

Set attributes such as byte cost, security, connection cost, and capacity for the adjacent link station with the LinkStaChar parameter. You can set any number of these options in any combination when entering the command. For more information on configuring this parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

For example, to define the characteristics of the link named "FINANCE3" for an effective capacity of 9600, a byte cost of 128, and a security value of SECURcnd, enter:

```
SETDefault -APPN LinkStaChar = FINANCE3 EffectCap=9600 ByteCost=128
      Security=SECURcnd
```



CAUTION: *If you change any of the default characteristics for a link to a network node, the characteristic must also be changed on the partner network node. For example, if you set the security level of the TG as GUarded on the local node, then you must also configure the security level as GUarded on the partner node. Otherwise, the characteristic will be valid in one direction only, from the local node to the partner node; the characteristic on the link in the opposite direction will not match.*

- 3 Repeat steps 1 and 2 for each network node that will establish direct connections (or links) with the local network node.

If you did not assign link names using the AdjLinkSta parameter, the system will assign them. To obtain a list of link names assigned, enter:

```
SHow -APPN LinkStaCONTROL
```

You can configure two or more links to the same node using parallel TGs. For more information on configuring parallel TGs, see "Configuring Parallel Transmission Groups" later in this chapter.

If you need to configure support for dependent LUs, proceed to the next section. If you do not need to do so, proceed to “Enabling the Network Node and Activating Links” later in this chapter.

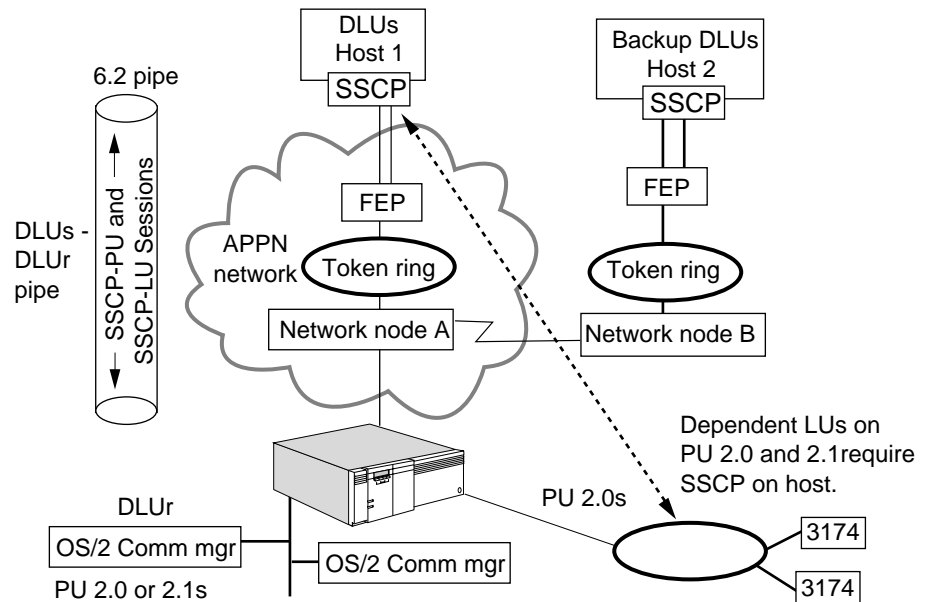
Configuring Dependent LU Support

Dependent logical unit support is required where you have PU type 2.0 or 2.1 nodes in the local network node's domain that will access a host via LU types dependent on the SSCP. LU types that are dependent on a Session Services Control Point (SSCP) are types 1, 2, 3, or type 6.2. Configuring dependent LU support on the network node enables the network node to act as a Dependent LU Requestor (DLUr) to enable a PU type 2.0 or 2.1 node to access the host, which acts as the Dependent LU Server (DLUs). You can have many PUs with dependent LUs accessing one primary DLUs and one backup DLUs.

PU type 2.0 nodes are nodes which do not have a control point. As a result, LUs on these nodes are “dependent” on SSCP services provided by the DLUs. PU type 2.1 nodes can have both independent and dependent LUs. The dependent LUs require the SSCP services from the host, while independent LUs do not.

Figure 154 is an example of PU type 2.0 and 2.1 nodes accessing a host DLUs with a bridge/router acting as the DLUr. In the configuration, the DLUs is *upstream* from the network node bridge/router, while the PU 2.0 and 2.1 nodes are *downstream* from the network node.

Figure 154 DLUr and DLUs Environment



This section is divided into two procedures:

- Defining your DLUs
- Configuring links to nodes requesting DLUr services

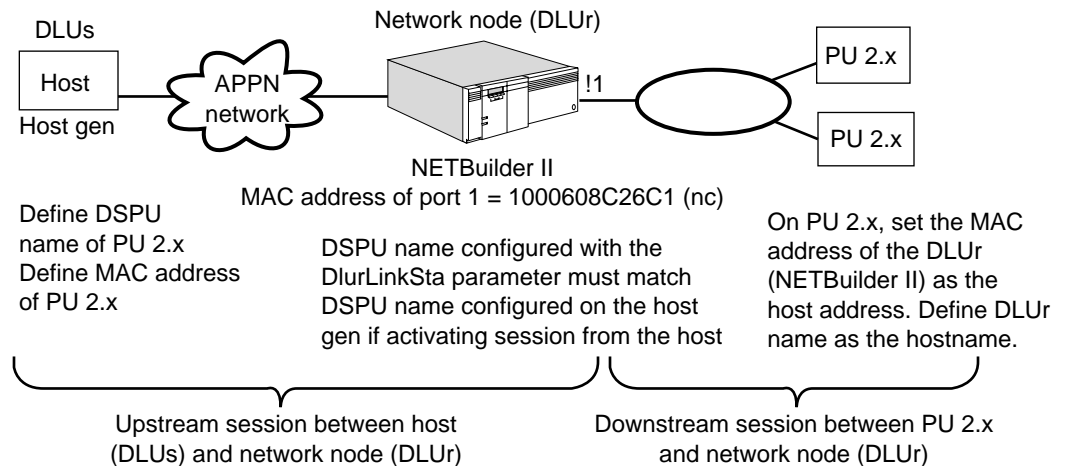


If the DLUs will be accessed over a WAN using Frame Relay, you also will need to configure the APPN Frame Relay interface.

Figure 155 is an example of a DLUr and DLUs configuration. In the configuration, the downstream physical unit (DSPU) defined in the host configuration must

match the DSPU name configured on the network node using the `DlurLinkSta` parameter. For the PU 2.x to access the host, the MAC address of the local node must be configured as the host address on the PU. The PU thinks the host address is for the remote host providing the service, but the network node address is used to establish the session to the network node. The network node then establishes the SSCP-LU and SSCP-PU sessions with the host.

Figure 155 DLUs and DLUr Configuration



Defining the Default DLUs and Backup DLUs

When you define the DLUs on the network node, you are configuring the default DLUs and backup DLUs that the local node (acting as the DLU requestor) will send the SSCP traffic to. The DLUs does not need to be directly connected to the local network node, and there can be multiple network nodes in between.

When a dependent LU makes a session request to the local network node for a dependent LU server, the local node tries to find the DLUs using the following hierarchy of steps:

- The system first looks for the DLUs assigned to the DLUr link station using the `DlurLinkSta` parameter (see “Defining Downstream Links to Nodes with Dependent LUs”).
- If that DLUs is unavailable or no DLUs was assigned to the DLUr link station, then the system tries to use the backup DLUs assigned using the `DlurLinkSta` parameter.
- If the backup DLUs is unavailable or no backup DLUs was assigned to the DLUr link station, then the local node tries the default DLUs configured using the `DlurDefaults` parameter.
- If the default DLUs is unavailable, then the local node tries the default backup DLUs configured using the `DlurDefaults` parameter.

To configure the default DLUs and backup DLUs, use:

```
SETDefault -APPN DlurDefaults [Dlus=( <name> | UNdef )
[Backup=( <name> | UNdef )]
```

This command specifies the default DLUs and the backup DLUs. You can configure one default DLUs and one default backup DLUs on the local network node.

For example, to configure a primary DLUs named "VTAM1" and a backup DLUs named "VTAM2," enter:

```
SETDefault -APPN D lurDefaults = DLUS=VTAM1 BACKUP=VTAM2
```

To change the name of a primary or backup DLUs, repeat the command and enter a different name. To remove the name of a primary or backup DLUs, enter the command but specify "UNdef." For example, to remove VTAM2 as the backup DLUs, enter:

```
SETDefault -APPN D lurDefaults = BACKUP=UNdef
```

Defining Upstream Links for Path to DLUs

You can have any number of intermediate network nodes in your APPN network between the local network node DLUr and the DLUs host. To define the upstream link for the path to the DLUs, you configure the upstream network node as a normal adjacent link station. No special configuration is required. The only requirement is that you must be able to establish 6.2 LU to LU sessions between the local network node DLUr and the DLUs host.

Defining Downstream Links to Nodes with Dependent LUs

If you have PU 2.0 nodes or PU 2.1 nodes with dependent LUs in the network node domain, then you must configure DLUr link stations to each of these nodes. Because these nodes function differently from normal APPN nodes, you cannot configure DLUr link stations and normal adjacent link stations to the same node. However, a node can have CP-CP sessions and still require DLUr. If that is the case, add these nodes using this procedure.

To add a link to PU 2.0 and 2.1 nodes that require DLUr services, follow these steps:

1 Select one of the following:

- a If you are running normal APPN traffic to and from DLUr link stations, define each DLUr link station using:

```
ADD !<port> -APPN D lurLinkSta <max_btu_size(256-8912)> <[Cmac | Ncmac] dest  
media addr> <dspu name> [Sap=<num>] [Nodeid=<ID>] [LinkName=<name>]  
[Dlus=<[netid.]name|UNdef>] [Backup=<[netid.]name|UNdef>]  
[TGprof=<name>] [AutoStart=(Yes|No)] [CPSess=(Yes|No)] [PU2=(Yes|No)]  
[HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

Using this command, you specify the maximum BTU size, the destination address of the DLUr link station, and the DSPU name of the PU 2.0 device. If the host will activate the session with the DLUr link station, then the DSPU name you configure here must match the name on the host configuration.

You also specify the primary DLUs and backup DLUs that the DLUr link station will access. If a primary and/or backup DLUs is not specified, then the default primary and backup DLUs configured using the D lurDefaults parameter will be used. The default for AutoStart is No. If you want the link to automatically be activated when the network node is enabled, specify AutoStart=Yes.

- b If you set the port DLC type to SDLC to run SDLC traffic to and from DLUr link stations, define each SDLC DLUr link station using:

```
ADD !<port> -APPN S dlcD lurLinkSta <max_btu_size>(265-8912) <station  
addr>(Hex 1-FE) <dspu name> [Nodeid=<ID>] [LinkName=<name>]  
[Dlus=[netid.]name] [Backup=[netid.]name] [TGprof=<name>]
```

```
[AutoStart=(Yes|No)] [PU2=(Yes|No)] [HPR=(Yes|No)]
[CPsEss=(Yes|No)] [SendWindow=<num>] [ContactTimer=<num>]
[NoRspTimer=<num>] [NoRspTimRetry=<num>]
```

Using this command, you specify the maximum BTU size, the destination address of the DLUR link station, and the DSPU name of the PU 2.0 device. If the host will activate the session with the DLUR link station, then the DSPU name you configure here must match the name on the host configuration.

You also specify the primary DLUs and backup DLUs that the DLUR link station will access. If a primary and/or backup DLUs is not specified, then the default primary and backup DLUs configured using the DLUrDefaults parameter will be used. You can also specify SDLC attributes such as the SendWindow, ContactTimer, NoRspTimer, and NoRspTimRetry values. For more information on these values, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.



APPN over SDLC connections is supported on all types of HSS 3-Port modules, including V.35, RS-232, and RS-449.

- 2 Repeat the previous step for each PU 2.0 or 2.1 node that will access a DLUs through the local network node.

Using VTAM Program Temporary Fixes

VTAM Program Temporary Fixes (PTFs) are required on a mainframe when APPN DLU services are used. Mainframe network management (NetView) services will not function for downstream physical units (PUs) if the PTFs are not installed. VTAM Version 4.2 requires PTF #UW20787. VTAM Version 4.3 requires PTF #UW20788.

Symptoms of this problem result from a lack of network management data for PUs that are downstream of a NETBuilder II using APPN DLU services. The NetView message "AAU251I AAUDRTIB 02 UNEXPECTED SENSE CODE X'1002' ENCOUNTERED FOR TARGET=pu_name" is printed in the log file when this problem occurs.

Enabling the Network Node and Activating Links

After you have set up the bridge/router as a network node and defined links to other network nodes you can now enable the network node and activate the links you defined in the previous sections.

To enable the network node and activate the links, follow these steps:

- 1 To enable the bridge/router to function as an APPN network node, enter:

```
SETDefault -APPN CONTROL = Enable
```

When you enable the APPN network node, you will receive a message similar to the following:

```
Wed Dec 31 16:11:15 1995 LOCAL NETWORK NODE US3COMHQ.GOLD IS STARTED
```

After the network node is enabled, the bridge/router can communicate with other APPN network nodes, and can accept incoming link requests from end nodes.

You can totally disable the network node, or you can dynamically disable the network node so that when you reboot the bridge/router, the network node automatically is re-enabled. For more information on disabling the network node, see "Disabling the Network Node" later in this chapter.

- 2 If you configured adjacent link stations and you set AutoStart to No or configured DLUr link stations and did not set AutoStart to Yes, activate these links using:

```
SET -APPN LinkStaCONTRol = <LinkName> Activate
```

Repeat this step for each of the links you defined in the previous sections. After you have enabled the network node and activated your basic links, the basic network node will be operating. Other network nodes will be able to initiate sessions with the local node and receive sessions from the local node. In addition, end nodes in the local node's domain will be able to initiate session requests with the network node.

For additional configuration, see "Customizing the APPN Router" later in this chapter.

Dynamic Configuration Options

After the network node is enabled, you can configure different options such as adjacent link stations, transmission group (TG) characteristics, and port characteristics. Depending on the task, you can configure these options without disabling the network node or disrupting sessions on ports or TGs not affected. Table 27 lists some of the APPN entities that you can and cannot dynamically configure while the network node is operating.

Table 27 APPN Dynamic Configuration Options

| Configuration Option | Parameter | Dynamic Configuration Allowed | Additional Information |
|---|--------------------------------------|-------------------------------|--|
| Predefine LEN end node LUs | AdjLenDef | Yes | |
| Add or delete adjacent link stations | AdjLinkSta | Yes | Port the link station is mapped to can be enabled while configuring. Must activate link using LinkStaCONTRol parameter to take effect. To delete link station, must deactivate it first. |
| Adjacent link station characteristics | LinkStaCHar | Yes | Cannot make changes if link is active. You must first deactivate the link and then reactivate it after making the change. |
| Create a customized class of service, and change node row and TG row values | ConfigCos
COSNodeRow
COSTgRow | Yes | See the Configuring APPN Class of Service chapter for more information. |
| Enable connection network | ConnNetwork
Def | Yes | The port the connection network is added to can be enabled when configuring. |
| Define a customized class of service to the system | CosDef | Yes | See the Configuring APPN Class of Service chapter for more information. |
| Add or delete directory entries | DirectoryEntry | Yes | |
| Set the default DLUs and backup DLUs | DlurDefaults | Yes | |
| Define DLUR link stations | DlurLinkSta | Yes | If the link is active, you cannot make changes. Deactivate the link before making changes. |
| Activate and deactivate link stations | LinkStaCONTRol | Yes | |
| Set the local node name and resistance | LocalNodeName
LocalNodeResistance | No | Must be configured before enabling the network node. |
| Map mode names to a class of service | ModetoCosMap | Yes | See the Configuring APPN Class of Service chapter for more information. |
| Change APPN port characteristics and define the APPN port | PortCHar
PortDef | Yes | If port is activated, must first deactivate the port using PortCONTRol parameter before changing characteristics or definitions. Port must then be reactivated after making changes. |
| Activate and deactivate APPN port | PortCONTRol | Yes | |
| Set queue priority | QueuePriority | Yes | See the Prioritizing Multiprotocol Data chapter for more information. |
| Add or delete adjacent SDLC link stations | SdlcAdjLinkSta | Yes | Port the link station is mapped to can be enabled while configuring. Must activate link using LinkStaCONTRol parameter to take effect. To delete link station, must deactivate it first. |
| Define SDLC DLUR link stations | SdlcDlurLinkSta | Yes | If the link is active, you cannot make changes. Deactivate the link before making changes. |

Configuring the APPN Router for Wide Area Networks

To configure your APPN router to perform routing over Frame Relay, see the Configuring Wide Area Networking Using Frame Relay chapter. APPN routing over SMDS and X.25 is not supported unless you are using DLSw. For information on routing over PPP connections, see the Configuring Wide Area Networking Using PPP chapter. For information on wide area networking using ISDN, see the Configuring Wide Area Networking Using ISDN chapter. For more information on data link switching, see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

APPN routing over ATM is not supported.

Verifying the APPN Router Configuration

To verify that the APPN router you configured is recognized by the APPN network and is receiving incoming session requests, follow these steps:

- 1 Display information on ports configured for APPN using:

```
SHow [!<port>] -APPN PortDef
```

- 2 Verify that the ports configured for APPN are active using:

```
SHow [!<port>] -APPN PortCONTRol
```

If a port is shown as "Not Defined" in the display, that indicates the port was not defined as an APPN port using the SETDefault !<port> -APPN PortDef command.

- 3 Verify the local node name assigned to the APPN router by entering:

```
SHow -APPN LocalNodeName
```

Note the local node name so you will recognize it in displays later in this procedure.

- 4 Verify the adjacent link stations the APPN router is linked to by entering one or both of the following commands:

```
SHow -APPN AdjLinkSta
```

```
SHow -APPN SdlcAdjLinkSta
```

For more information about this display, see "Adjacent Link Station Information" later in this chapter.

- 5 Verify whether links to adjacent link stations and DLUR link stations are active by entering:

```
SHow -APPN LinkStaCONTRol
```

For more information about this display, see "Current Status of Link Stations" later in this chapter.

- 6 Verify information for all adjacent network nodes the APPN router can communicate with by entering:

```
SHow -APPN NNTopology
```

For more information about this display, see "Network Topology Information" later in this chapter.

- 7 Verify information for the number and status of all adjacent nodes the APPN router is communicating with by entering:

```
SHow -APPN AdjNodeStatus
```

The display shows the number of adjacent nodes, including adjacent nodes, and end nodes in the network node's domain, and characteristics for those nodes.

- 8 Verify that the APPN router is sending and receiving connections to other nodes and the status of those connections by entering:

SHoW -APPN CONNectiOn

For more information about this display, see “Active APPN Connections” later in this chapter.
- 9 Verify that LUs on other nodes are getting registered into the local node’s directory by entering:

SHoW -APPN DIRectory

For more information about this display, see “APPN Directory Information” later in this chapter.
- 10 Verify that the APPN router is handling intermediate session routing, and verify the status of any ISR sessions by entering:

SHoW -APPN ISRsessions

For more information about this display, see “Intermediate Session Routing Information” later in this chapter.
- 11 To display the status of all DLU servers that the local node has 6.2 sessions with, enter:

SHoW -APPN DluSStatus
- 12 To display a list of DLUR link stations, enter one or both of the following commands:

SHoW -APPN DlurLinkSta
SHoW -APPN SdlcDlurLinkSta
- 13 To display a list of downstream PUs, enter:

SHoW -APPN DluRStaus
- 14 To display a list of downstream LUs, enter:

SHoW -APPN DownStreamLU
- 15 Verify link activity for the node by entering:

SHoW -APPN AppnLOG

Troubleshooting the APPN Router

If the APPN router is not properly communicating with other nodes in the network, review the following procedure. For more information regarding APPN Service parameters, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

You can troubleshoot problems on an APPN network by following one or more of these steps:

- 1 Show the version of the software by entering:

SHoW -SYS VERSion
- 2 Show the path configuration by entering:

SHoW -PAth CONFIguration
- 3 Show the port configuration by entering:

SHoW -PORt CONFIguration
- 4 Show the system configuration by entering:

SHoW -SYS CONFIguration

- 5 Show the APPN configuration by entering:
SHoW -APPN CONFIguration
- 6 Check the status of APPN ports by entering:
SHoW -APPN PortCONTRol
- 7 Check the status of adjacent link stations by entering:
SHoW -APPN LinkStaCONTRol
- 8 Check the status of active connections by entering:
SHoW -APPN CONNectiOn ALL
- 9 Check the status of adjacent nodes by entering:
SHoW -APPN AdjNodeStatus
- 10 Check the status of ISR sessions by entering:
SHoW -APPN ISRsessions
- 11 Check the status of transmission groups by entering:
SHoW -APPN TG ALL
- 12 If you cannot reach a specific LU in the APPN network, determine if a route exists between the local node and the LU using:

APpnPING [netid.]<partner_lu_name> [Mode=modename] [Size=N] [Consec=N] [Iterations=N] [Echo=Yes|No] [Userid=<string> [Password=<string>]]

The APpnPING command performs an APPC Ping to the other LU in the network. For more information on using the APpnPING command, see the Commands chapter in *Reference for Enterprise OS Software*.
- 13 Check the current status of LLC2 sessions by entering:
SHoW -LLC2 SESSions
- 14 Check the current statistics for LLC2 sessions by entering:
SHoW -SYS STATistics -LLC2
- 15 Perform an analyzer trace on the LLC2 LAN links.
- 16 Perform an analyzer trace on the PPP WAN links.

Customizing the APPN Router

After you have configured the local network node, the network node will operate as an APPN router, communicating with other network nodes and accepting incoming session requests from end nodes. You can customize the APPN router for greater control and security by performing the following tasks:

- Statically defining links (adjacent link stations) to end nodes
- Statically defining entries into the network node's directory

You also can customize the APPN router by configuring the following items:

- Links to connection networks
- Parallel TGs
- Data link switching (DLSw) between nodes
- APPN and Boundary Routing environments

Defining Links to End Nodes

You normally do not have to define links (adjacent link stations) to end nodes. In APPN, end nodes make a link request to a network node to access the network. When a network node provides routing and topology services for an end node, the network node is called the *network node server* for the end node. End nodes can have links to more than one network node at a time, but only one network node can be the network node server to that end node at one time.

Because end nodes make incoming link requests to the network node, the process is dynamic, meaning end nodes can link to one network node for a certain time, then break the link and link to another network node for a different session request. As a result, it may not be practical to statically define links to end nodes if you have different network nodes that can serve as network node servers. If you have many end nodes, statically defining links for each one may not be practical.

You may want to statically define links to end nodes if you have a secure environment or want greater control over the network.

The procedure to define links to end nodes in your network node domain is similar to the procedure used to define links to other network nodes. Low-entry networking (LEN) end nodes are a subset of end nodes, and you define links to LEN end nodes the same way. However, if the LEN end node has more than one LU, then you need to statically predefine these LUs; for more information, see “Preconfiguring LEN End Node LUs” later in this chapter.

To define links to end nodes in your network node domain, follow these steps:

- 1 Define the link to an end node on a port and specify the node type as EN using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn) <max_btu_size>(99-8912)
  [[Cmac|Ncmac] dest media addr] [Sap=<num>] [CPName=[netid.]cpname]
  [Nodeid=<ID>] [LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
  [CPSess=(Yes|No)] [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

or, if running SDLC traffic on the port:

```
ADD !<port> -APPN SdlcAdjLinkSta <type>(NN|EN|Learn)
  <max_btu_size>(99-8912) <station addr>(Hex 1-FE) [CPName=[netid.]cpname]
  [Nodeid=<ID>] [LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
  [CPSess=(Yes|No)][HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
  [SendWindow=<num>] [ContactTimer=<num>] [NoRspTimer=<num>]
  [NoRspTimRetry=<num>]
```



APPN over SDLC connections is supported on all types of HSS 3-Port modules, including V.35, RS-232, and RS-449.

In addition to the adjacent link station's node type, you specify the maximum BTU size, and the destination media address control (MAC) address for non-SDLC traffic, or the destination station address for SDLC traffic. Optionally, you can set the node's CP name, node ID, link name, TG profile, whether auto startup will be supported, and whether the link will support CP-CP sessions with the adjacent node. The default for end nodes is to support CP-CP sessions. For non-SDLC traffic, you can set the node's Service Access Point (SAP) number. For SDLC traffic, you can set SDLC attributes such as SendWindow, ContactTimer, NoRspTimer and NoRspTimRetry. If the adjacent link station will not support HPR, make sure to specify HPR=No to turn off HPR support.

For example, to add a link to an end node in an ISR network named “ENGREEN” to port 3 with a maximum BTU size of 1033 (specifying the appropriate MAC

address and not fully qualified CP name), and to specify the link will support auto startup, enter:

```
ADD !3 -APPN AdjLinkSta EN 1033 N100040C08ACE Sap=08 CPName=ENGREEN
      AutoStart=Yes HPR=No
```

For information on how to obtain the MAC address of a node, see the documentation for the end node device or applications. Most SNA and token ring environments use noncanonical MAC address formats. To convert a MAC address to canonical format, use the MacAddressConvert command.



If you set the -SYS MacAddrFmt parameter to noncanonical, then you do not need to precede the MAC address with N or Ncmac.

- 2 After you have defined the link to the end node, define the link characteristics using:

```
SETDefault -APPN LinkStaChar = <LinkStation name>
  [EffectCap=<string>] [ConnectCost=<0-255>] [ByteCost=<0-255>]
  [Security=<string>] [PropDelay=<string>] [Usd1=<0-255>]
  [Usd2=<0-255>] [Usd3=<0-255>]
```

With this command, you set attributes such as byte cost, security, connection cost, and effective capacity for the adjacent link station. For more information on configuring this parameter, see the description of the LinkStaChar parameter in the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

- 3 Repeat steps 1 and 2 for each end node (or LEN end node) that you will allow to link directly with the local network node.

Defining Links to Unknown Node Types

You may not know if a node is a network node or an end node, or know the node name or CP name. To define an adjacent link station to an unknown type of node, enter the ADD -APPN AdjLinkSta command or the ADD -APPN SdlcAdjLinkSta command and specify the node type as LEARN. If you specify LEARN, the system learns the node type as well as other information such as the node name and CP name. To add a link station to a node whose node type is learned, you must at least know the MAC address of the node. To add an SDLC link station to a node whose type is learned, you must at least know the station address of the node.

For example, to define a link station on port 4 to an unknown node type with a maximum BTU size of 1033 and a noncanonical MAC address of %100040C08ACE, enter:

```
ADD !4 -APPN AdjLinkSta LEARN 1033 %100040C08ACE
```

Defining Entries in the Network Node's Directory

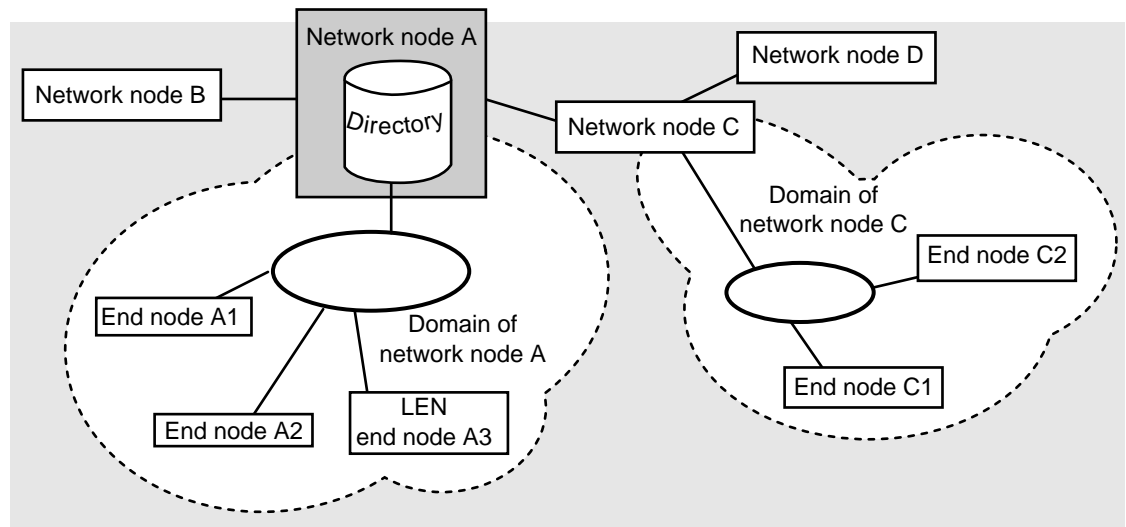
The network node maintains a directory of nodes it knows about, and any logical units on those nodes. When an incoming session request comes to the network node, the network node uses the information stored in the directory to determine the location of the destination LU. If the destination LU is not located in the network node's domain, the network node sends locate requests to adjacent network nodes.

Figure 156 shows a network node and what nodes would be included in the network node's directory. Network node A is the local node. The shaded area indicates nodes that would be included in network node A's directory, either dynamically learned or statically defined. End nodes A1 and A2 are in network node A's domain, and would be dynamically learned and added to the directory. LEN end node A3 is also in network node A's domain; if there are other LUs on

that LEN node other than the LU for the node's CP, these additional LUs would have to be statically defined (for more information on defining LEN end node LUs, see below). Network nodes B and C are also dynamically learned in network node A's directory because both are adjacent nodes, one hop away.

Network node D would not be included in the directory because it is not an adjacent node, and is two hops away. End nodes C1 and C2 would not be included because they reside in network node C's domain; as a result, end nodes C1 and C2 would be included in network node C's directory.

Figure 156 Nodes Included in the Network Node Directory (Example)



When you display the directory, the display shows the location of the logical units. In this example for network node A's directory, end nodes LUs on A1 and A2 would be *registered* entries, meaning they were dynamically learned, and the location would be *domain*, meaning they reside in the local domain. The LU on LEN end node A3 would be a *home* entry (meaning it was statically defined), and the location would be *domain*. For more information on displaying the directory, see the DIRectory parameter in the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

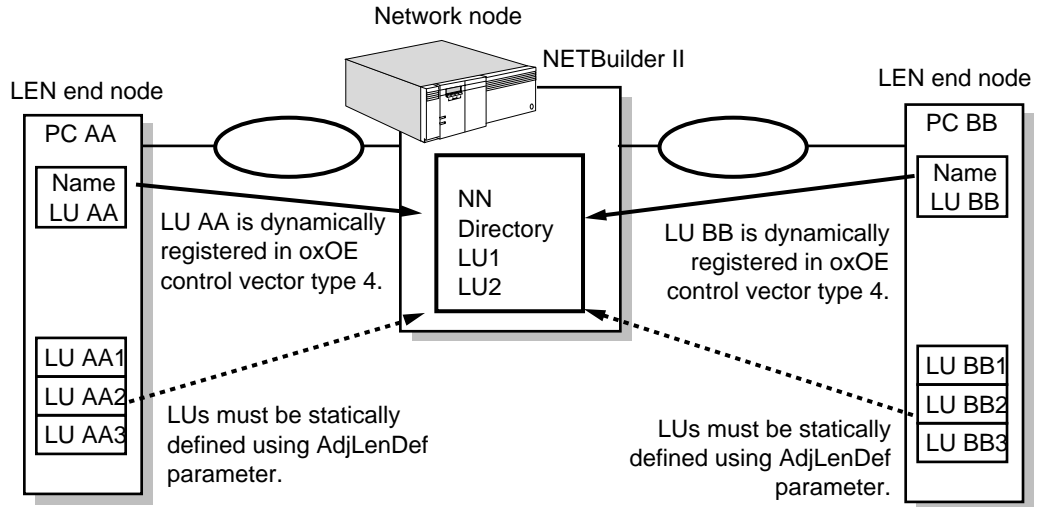
Preconfiguring LEN End Node LUs

When a LEN end node is added to an APPN network as an adjacent link station, the LEN end node sends an XID3 to the network node when the link activates. In this XID3, the LEN end node's CP name is sent in the ox0E control vector type F4. This CP name maps to the LEN end node's LU name. However, if the LEN end node has more than one LU, then you must statically preconfigure those LUs into the network node directory.

Figure 157 is an example of two LEN end nodes connected directly with the intermediate network node. On a LEN end node, the single LU that maps to the node's CP name that was sent in the control vector is dynamically registered

through the XID3 with the network node when the link is activated. In the figure, both LU AA on PC AA and LU BB on PC BB would be dynamically registered.

Figure 157 LEN End Node LU Registration



You must statically define LEN end node LUs in the following situations:

- If the LU name does not match the CP name
- If the control vector does not send the LU name
- If the LEN end node has LUs in addition to the LU registered through the XID3

The PCs in the figure show the last situation, in which both PCs have additional LUs. Since the XID3 only registers the LU for the network name control vector, these additional LUs must be statically defined into the network node's directory using the AdjLenDef parameter.

In APPN, when two LEN end nodes have a peer-to-peer connection, either side can activate the connection or start a session to the other node. The LEN end node that activates the connection sends a BIND to the other node. For the connection to work, the LEN end node that receives the BIND has to be preconfigured into the network node directory so that the network node can find the destination LU to send the session request.

For example, if LU AA in the figure activates a session to LU BB2, then LU BB2 must be preconfigured in the network node's directory; otherwise, the session request will not be successful. If LU AA2 wants to activate a session to LU BB2, both LUs need to be preconfigured in the network node directory. After these two LUs are preconfigured, either LU can initiate a connection. Also, once LUs are preconfigured in the network node directory, other LUs in the network can find

them. Conversely, if LUs that require preconfiguration are not in the network node directory, other LUs in the network will not find them.

To statically define LEN end node LUs into the network node directory, follow these steps:

- 1 If you have LEN end nodes with more than one LU, or LEN end nodes in the network node domain that will receive BINDs that do not match the CP name in the XID, you must statically define these LUs using:

```
ADD -APPN AdjLenDef [adjnetid.]<adjcpname> [adjlu ...]
```

This command statically defines any logical units on the LEN end node in the local network node server's directory.

For example, to add the three LUs named AA1, AA2, and AA3 on the LEN end node AA, enter:

```
ADD -APPN AdjLenDef AA AA1 AA2 AA3
```

When you add CP and LU names, the names are converted to all uppercase, even if you enter some lowercase letters. When entering this command, you can use the not fully qualified CP name. Use this command to define up to 4 LUs at a time; to define additional LUs, reenter the command. You can register up to 256 LUs on the network node.

- 2 Repeat the previous step for each LEN end node in your network node's domain with more than one LU. The entries take effect immediately.

For information on how to display entries in the directory, see "APPN Directory Information" later in this chapter.

Deleting LEN End Node LUs

You can delete statically defined LEN end node LU entries from the directory using the `DELeTe -APPN AdjLenDef` command. You can specify individual LUs to be deleted. If you do not specify LU names in the command, the entire adjacent node is deleted from the directory, along with all LUs belonging to the adjacent node.

For example, to delete the LUs named AA2 and AA3 on node AA, enter:

```
DELeTe -APPN AdjLenDef AA AA2 AA3
```

Adding Entries

In most configurations, you do not need to statically define network nodes and regular end nodes in the directory. You do need to determine how many cached entries you will allow and if you have LEN end nodes that receive BINDs, you must statically define them for the directory.

Unlike LUs on LEN end nodes that may require static definition, LUs on end nodes and network nodes are normally learned dynamically. Although not required, you can also statically predefine the location of LUs on other nodes in the network.

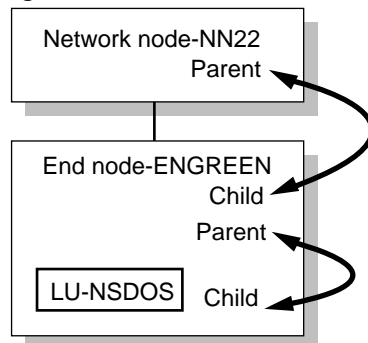
To preload entries into the APPN directory cache, use:

```
ADD -APPN DirectoryEntry [netid.]<resource name>
  <type(LU|EN|NN|Wild)> [[netid.]<parent_name> <parent_type(EN|NN)>]
  [[netid.]<grandparent_name> <grandparent_type(NN)>]
```

Using this command, you enter the resource type into the directory. If the resource is not a network node, you must specify the parent name and parent type of the resource. The resource parent and child is used for destination node broadcast searches. When a node or LU is a child resource, the child must reply to the parent for a search to be completed.

Figure 158 is a simple example of how this directory hierarchy works. In this example on the network HQ, the network node NN22 is the parent resource to the end node ENGREEN, which is the child. ENGREEN is the parent resource to the LU named NSDOS, which is a child resource residing on that end node.

Figure 158 Parent and Child Directory Entries



To add a directory entry in which the LU named HQ.NSDOS is the child to the end node HQ.ENGREEN, which is a child entry to the network node HQ.NN22, enter:

```
ADD -APPN DirectoryEntry HQ.NSDOS LU HQ.ENGREEN EN HQ.NN22 NN
```

In this example, the network node HQ.NN2 is the grandparent entry to the LU HQ.NSDOS. When entering an entry for a grandchild (three levels down), you must specify the grandparent name. The grandparent type will always be a network node.

Alternatively, you can enter these directory entries separately. For example, you can enter the following three commands, the first to define the network node, the second to define a child entry for the end node, and the third to define a child entry for the LU:

```
ADD -APPN DirectoryEntry HQ.NN22 NN
ADD -APPN DirectoryEntry HQ.ENGREEN EN HQ.NN22 NN
ADD -APPN DirectoryEntry HQ.NSDOS LU HQ.ENGREEN EN HQ.NN22 NN
```

You can add wildcard entries to the directory. Wildcards are of two types: full, where you just enter an asterisk (*), or partial, where you enter part of the name and an asterisk (for example, LU7*).

To add a partial wildcard entry for all LUs that start with "LU7" as child entries to HQ.NN22, enter:

```
ADD -APPN DirectoryEntry LU7* Wild HQ.NN22 NN
```

Deleting Entries

To delete entries from the network node directory, use:

```
DELeTe -APPN DirectoryEntry [netid.]<lu_name> <type(LU|EN|NN|Wild)>
```


For example, to delete the directory entry NSDOS, for the LU on ENGREEN, enter the following command, entering the LU name and specifying the type as LU:

```
DELEte -APPN DirectoryEntry HQ.NSDOS LU
```

If you delete a resource, all the child entries and grandchild entries belonging to that resource will also be deleted. For example, if you delete the grandparent entry HQ.NN22, the child entry HQ.ENGREEN and the grandchild entry HQ.NSDOS will also be deleted.

Configuring Parallel Transmission Groups

A transmission group (TG) is the link between two nodes. By configuring parallel TGs, you can configure two links from the local network node to the same adjacent node. This can provide more flexibility in routing APPN traffic to and from a single device. With parallel TGs, you can configure two links between the same two nodes, but not more than two.

Parallel TGs are not recommended for links over the same LAN, because there is no practical benefit for doing so; if you have parallel TGs over the same LAN and the LAN is busy, then both TGs will be busy.



If you configure parallel TGs between two NETBuilder II network nodes, then you only need to configure the partner node as an adjacent link station on one side.

There are several reasons why parallel TGs can be useful on your network:

- They can provide redundant links between nodes, to enable one link to take over if the other fails.
- You can assign different security levels to different TGs between nodes, allowing greater control over the traffic.
- You can assign different classes of service to each of the two TGs, allowing you to isolate different types of traffic over each link.
- You can have greater bandwidth between two nodes

When running parallel TGs, the CP-CP sessions can only go over one link at a time. With CP-CP session error recovery, if the link goes down the CP-CP sessions can be brought back up on the other link. For more information, see “CP-CP Sessions on Parallel TGs” later in this chapter.

Figure 159 is an example of parallel TGs being used for redundant links. In the configuration, both links between the network nodes are running at the same speed, and are running the same type of traffic. Each link is over a different port.

Although the links are redundant, if one link fails the traffic is not automatically switched to the second link. Unlike connectionless protocols, which can automatically switch links if a link fails, APPN is connection-oriented. As a result, if a link fails you will lose data, but you can restart your sessions over the second link.

Figure 159 Parallel TGs for Redundant Links

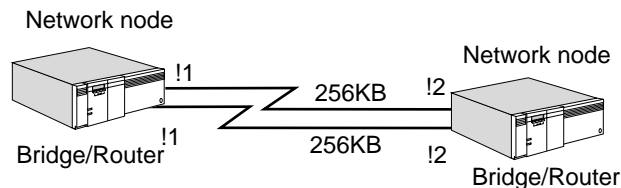


Figure 160 is an example of parallel TGs being sent over two different LANs. This configuration allows you to have redundancy between two nodes in your LAN environment. If one LAN fails, then you can restart sessions over the second LAN. If you configured both links on the same LAN, and the LAN fails, then both nodes would be isolated.

Figure 160 Parallel TGs over Different LANs for Resiliency

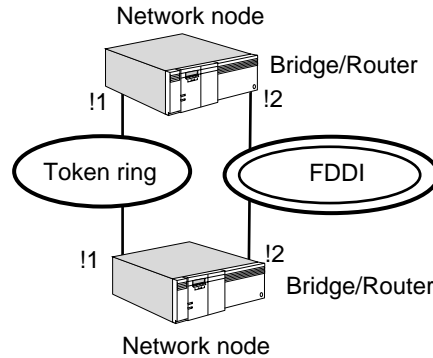
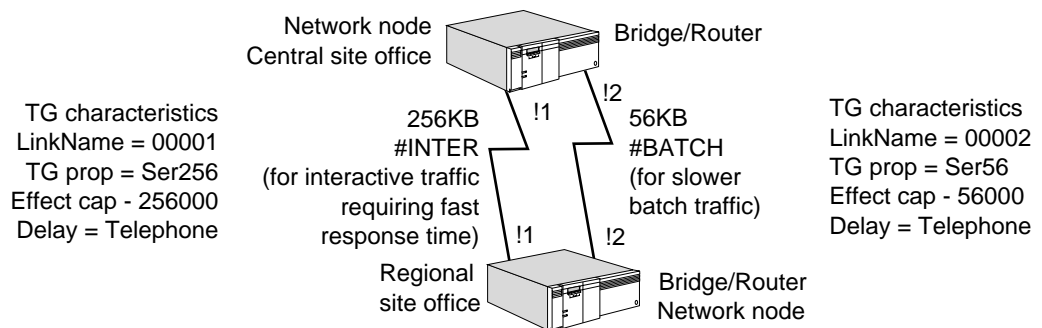


Figure 161 is a configuration in which parallel TGs are being used to isolate different types of traffic through different classes of service. The link on the left is set for a capacity of 256 KB and is being used for interactive traffic between terminals and the host; this type of traffic demands quicker response time so the class of service (COS) being used allows for a higher priority and the link is set for a higher speed. The link on the right is being used for lower speed batch transmissions, and as a result, is using the BATCH class of service and the link is set to a lower speed. In this example, the interactive traffic will be prioritized higher than the batch traffic.

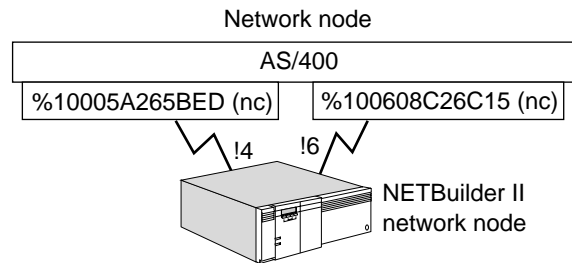
For more information on configuring APPN class of service, see the Configuring APPN Class of Service chapter.

Figure 161 Parallel TGs for Isolating Class of Service Traffic



Configuring Parallel TGs on the Network Node

Figure 162 is an example of a NETBuilder II bridge/router network node with parallel TGs over two different ports to an AS/400. In this example, the TGs are on two different FDDI rings, one being used for primary traffic and the other used as a backup.

Figure 162 Configuring Parallel TGs

To configure the parallel TGs for ports 4 and 6 on the NETBuilder II bridge/router in the figure, follow these steps:

1 Define the ports using:

```
SETDefault !<port> -APPN PortDef = <DLC type>
(LLC2|FR|PPP|DLSW|SDLC|UNdef) <max_btu_size>(99-8192)
[ActLimit=<limit>(1-512)] [TGprof=<name>] [HPR=(Yes|No)]
[ErrorRecovery=(Yes|No)] [DatMode=(Half|Full)] [ROle=(Pri|Sec|Neg)]
```

Define port 4 for LLC2 traffic, a maximum BTU size of 1033, and assign the TG profile "FDDI" by entering:

```
SETDefault !4 -APPN PortDef = LLC2 1033 TGprof=FDDI
```

Define port 6 for LLC2 traffic, a maximum BTU size of 1033, and assign the TG profile "FDDI" by entering:

```
SETDefault !6 -APPN PortDef = LLC2 1033 TGprof=FDDI
```

2 Define the adjacent link stations for both ports using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn) <max_btu_size>(99-8912)
[[Cmac|Ncmac] dest media addr] [Sap=<num>] [CPName=[netid.]cpname]
[Nodeid=<ID>] [LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
[CPSess=(Yes|No)] [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

Add the adjacent link station to port 4 to the destination media address on the AS/400 (entering the address in noncanonical format), a SAP of 08, and to specify autostart and CP-CP session activation by entering:

```
ADD !4 -APPN AdjLinkSta NN 1033 %10005A265BED Sap=08 TGprof=FDDI
AutoStart=Yes CPSess=Yes
```



CP-CP sessions can only be active over one TG at a time.

Add the adjacent link station to port 6 for the different destination media address, a SAP of 08, specifying autostart and support for CP-CP sessions by entering:

```
ADD !6 -APPN AdjLinkSta NN 1033 %100608C26C15 Sap=08 TGprof=FDDI
AutoStart=Yes CPSess=Yes
```

You can configure any adjacent link station characteristics using the LinkStaChar parameter.

You cannot assign specific numbers to specific TGs. The TG numbers are assigned through negotiation between the two nodes.

You can also configure parallel TGs for links to an SDLC device. You perform the same procedure, but you use the SdlcAdjLinkSta parameter.

CP-CP Sessions on Parallel TGs

When parallel TGs are configured between 3Com network nodes and both TGs support CP-CP sessions, a CP-CP session on one TG will not switch to the other TG if the user disables the port or path. This situation occurs because both sides learn about the link failure at different times. The network node with the disabled port or path learns about the link failure immediately and tries to bring CP-CP sessions up on the second TG. However, the second network node does not learn about the link failure until LLC2 times out. Because the node thinks the link is still up, the second network node does not allow CP-CP sessions to start on the second TG. After five attempts at bringing up CP-CP sessions on the second TG, the second TG will be flagged as not supporting CP-CP sessions, which prevents CP-CP sessions from coming up on that second TG.

To prevent this situation, manually stop the first TG by entering the SET -APPN LinkStaCONTRol <LinkName> Deactivate command before disabling the port and path. By doing this, both network nodes then learn that the link has gone down at the same time, and CP-CP session can be activated on the second TG.

Parallel TGs and Source Route Dual-TIC Topologies

You can configure parallel TGs in environments in which dual or multiple token ring interface cards (TICs) are configured on front-end-processors. For more information on dual-TIC topologies, see "Configuring DLSw for Dual-TIC Topologies" in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

Configuring DLSw Between Network Nodes

You can configure your APPN network so that you can send SNA traffic encapsulated in TCP packets over an IP network between two APPN network nodes using DLSw.

To configure DLSw between APPN nodes, additional configuration is necessary. Figure 163 is an example of two bridge/routers acting as APPN network nodes using DLSw to encapsulate SNA traffic in TCP packets across an IP internetwork. Table 28 lists the commands that need to be configured on each bridge/router in the figure.

Figure 163 Configuring DLSw Between Two APPN Network Nodes

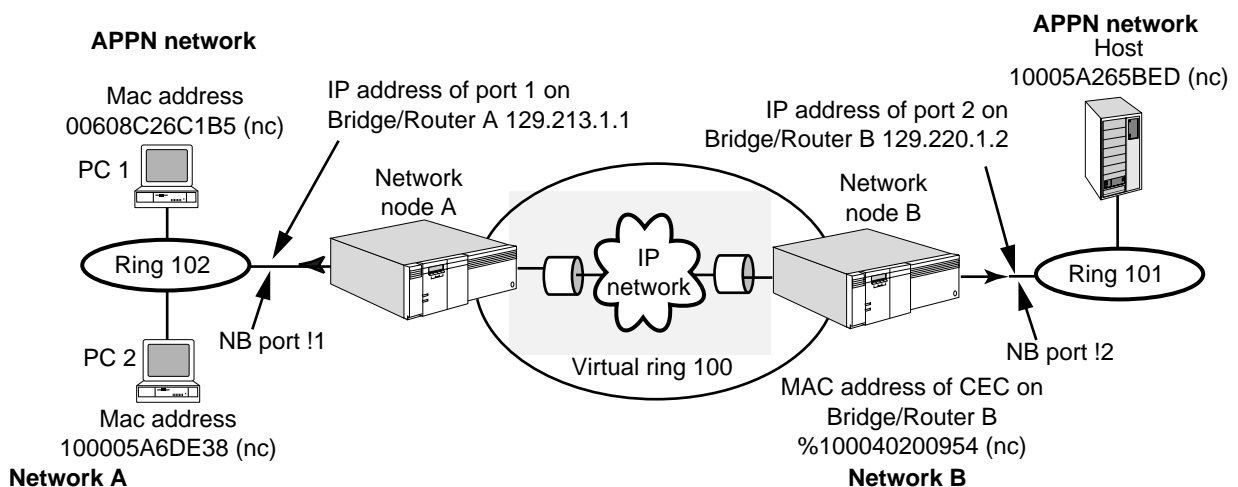


Table 28 Commands to Configure DLSw Between Two APPN Network Nodes

| Commands entered on Bridge/Router A | Commands entered on Bridge/Router B |
|---|---|
| SETDefault -IP CONTrol = Enable | SETDefault -IP CONTrol = Enable |
| SETDefault -TCP CONTrol = KeepAlive | SETDefault -TCP CONTrol = KeepAlive |
| SETDefault -TCP KeepAliveLimit = 3 | SETDefault -TCP KeepAliveLimit = 3 |
| SETDefault !1 -LLC2 CONTrol = Enable | SETDefault !2 -LLC2 CONTrol = Enable |
| SETDefault -LLC2 TUNnelVRing = 100 | SETDefault -LLC2 TUNnelVRing = 100 |
| SETDefault -DLSW Interface = 129.213.1.1 | SETDefault -DLSW Interface = 129.220.1.2 |
| ADD !1 -DLSW PEer 129.220.1.2 | ADD !1 -DLSW PEer 129.220.1.1 |
| SETDefault -DLSW CONTrol = EnableSNA,
DisableNetBios | SETDefault -DLSW CONTrol = EnableSNA,
DisableNetBios |
| SETDefault !0 -APPN PortDef = DLSW 1033 | SETDefault !0 -APPN PortDef = DLSW 1033 |
| ADD !0 -APPN AdjLinkStation NN 1033
N%100040200954 | |

As shown in the figure, you configure the network nodes as DLSw peers and the DLSw tunnel interface information using the normal procedure. For specific instructions on how to configure DLSw peers, see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.



You cannot perform bridging and tunneling of the same MAC address from an end station. You can perform either bridging only or tunneling only, but not both at the same time.

After configuring the two bridge/routers as DLSw peers, to configure DLSw tunneling between two APPN network nodes, follow these steps:

- 1 On both APPN network nodes acting as DLSw tunnel peers, configure the APPN port definition using the SETDefault !<port> -APPN PortDef syntax, specifying DLSw as the DLC type.

When specifying the port definitions for DLSw, you must specify the port number as !0. You only need to set the port definition for !0 for ports used for DLSw, and you should not specify !0 when setting the port definition for any other DLC type.

On bridge/router A in the figure, using port 0 and setting a maximum BTU size of 1033, enter:

```
SETDefault !0 -APPN PortDef = DLSw 1033
```

On bridge/router B in the figure, using port 0 and setting a maximum BTU size of 1033, enter:

```
SETDefault !0 -APPN PortDef = DLSw 1033
```

The maximum BTU size does not have to match on both sides of the tunnel. If the maximum BTU sizes differ, the smaller value will be used.

- 2 On the bridge/router that will initiate the connection, configure the tunnel peer bridge/router as an adjacent link station using:

```
ADD !<port> -APPN AdjLinkSta <type> (NN|EN|Learn)
  <max_btu_size> (99-8912) [[Cmac|Ncmac] dest media addr] [Sap=<num>]
  [CPName=[netid.]cpname] [Nodeid=<ID>] [LinkName=<name>]
  [TGprof=<name>] [AutoStart=(Yes|No)] [CPSess=(Yes|No)]
```

When you enter the command, you specify that the peer is a network node, the maximum BTU size, and the MAC address of the tunnel peer. In this case, the tunnel peer will always be a network node, since the bridge/router can only serve as a network node. The MAC address you enter is the address of the tunnel peer bridge/router, not the destination SNA host (also shown in the figure).

In the example shown in the figure, enter the following command on bridge/router A to add the link station as a network node with a maximum BTU size of 1033 and a SAP value of 08:

```
ADD !0 -APPN AdjLinkSta NN 1033 N%100040200954 Sap=08
```

When adding the adjacent link station for DLSw, you must specify the port number as !0 to map to port 0 configured in the previous step.

For more information about configuring data link switching, see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter. For information on parameters in the DLSw Service, see the DLSw Service Parameters chapter in *Reference for Enterprise OS Software*.

Configuring APPN for Boundary Routing

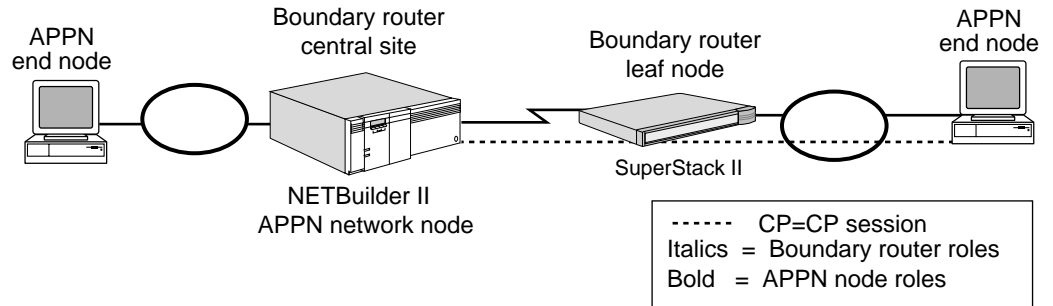
Boundary Routing is the 3Com system architecture that allows a network administrator to connect a central office network to a large number of small remote office networks (leaf networks). You can configure APPN to work in Boundary Routing environments, but there are limitations as to the types of configurations that can be set up. No additional APPN configuration is required for Boundary Routing environments. For information on Boundary Routing concepts and how to configure the central office router, see the Configuring Boundary Routing System Architecture chapter.

The 3Com Boundary Routing architecture is different from the APPN concepts of boundary nodes and border nodes. The 3Com APPN implementation supports the concept of boundary nodes but does not support the Systems Network Architecture (SNA) concept of border nodes or perform the border function. For clarification of these terms, see the IBM document, *APPN Architecture and Product Implementations Tutorial* listed in "IBM APPN References" later in this chapter.

Figure 164 is an example of a NETBuilder II bridge/router acting as an APPN network node and performing as the central site router in a Boundary Routing configuration connected to a SuperStack II NETBuilder bridge/router acting as a leaf node, which in turn is connected to a token ring network with APPN end nodes. The CP-CP session takes place between the NETBuilder II network node and the end node. The SuperStack II bridge/router acting as the leaf node does not participate in the CP-CP session, and cannot serve as an APPN node because the

APPN software is not supported on the SuperStack II bridge/router platform. In this situation, no special configuration is required on the SuperStack II bridge/router.

Figure 164 Configuring APPN with the Boundary Routing Architecture



You can also use Boundary Routing with APPN connection networks. For more information, see “Using Connection Networks in Boundary Routing Environments” later in this chapter.

If you are configuring APPN for Boundary Routing, the following special configuration is required on the central site bridge/router:

- You must configure the port definition to DLSw by entering:


```
SETDefault !0 -APPN PortDef = DLSW
```
- To enable the central site bridge/router to send ring information to the leaf node, you must configure the central site WAN link as a source route link, and turn on route discovery for both IP and LLC2 using:


```
SETDefault !<port> -SR RouteDiscovery = IP, LLC2
```

Configuring APPN Connection Networks

Connection networks are a way to provide greater scalability for growing APPN networks without exponentially increasing the number of broadcast traffic and overhead that could affect network performance. By configuring connection networks, you can enable links from one node to another through a virtual routing node; although the virtual routing node is an intermediate node, the link from the source node to the destination node is a virtual link.

Defining connection networks through virtual routing nodes is a method for setting up the network topology so that you can increase the number of nodes without flooding the network with topology data unit (TDU) broadcasts.

The following sections describe these connection network topics:

- How connection networks can be used to scale large APPN networks
- How to configure links to connection networks
- How to use connection networks in boundary routing environments

Using Connection Networks to Scale Larger Networks

Every time you add a new network node to the APPN network, you increase the amount of traffic overhead since each node broadcasts TDU updates to other directly connected nodes when the network changes. As you add nodes and scale the network, the network will be subject to increasing numbers of broadcasts, including Locate broadcasts, reducing network performance.

Figure 165 is an example of a fully meshed APPN network in which all eight nodes are directly connected to each other. Although each node is one hop from each

other, the large number of TGs means an exponential number of TDU broadcast updates flooding the network.

Figure 165 Meshed APPN Network without Virtual Routing Nodes (Direct Links)

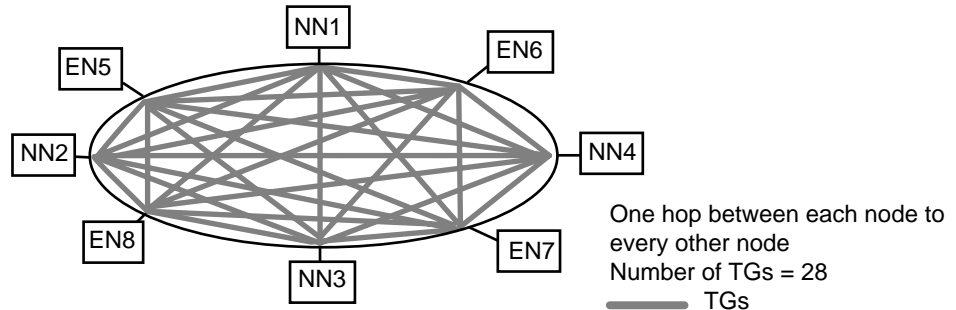
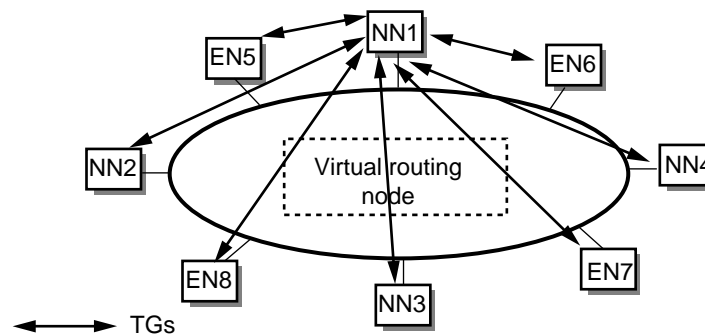
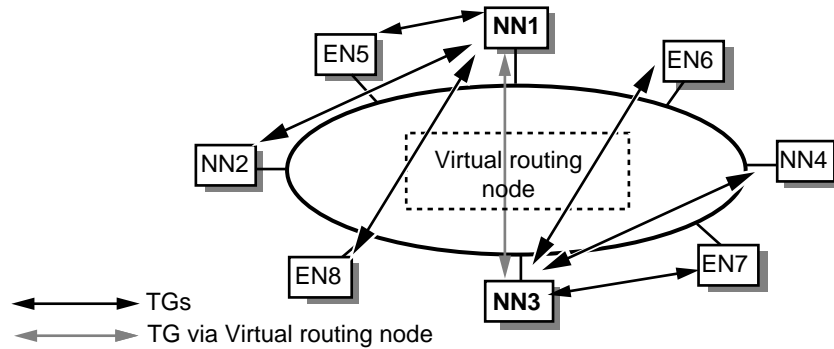


Figure 166 shows the same network, but with a virtual routing node being used to provide any-to-any connectivity between each node. In this configuration, NN1 is the focal point through which all links go through. Each node only requires link definitions to the common network node (NN1), and the virtual routing node. Because of the virtual routing node, the session data is not routed through real network nodes, reducing the number of CP-CP sessions as well as the number of TDU updates and Locate broadcasts.

Figure 166 Network with Virtual Routing Node (One Point of Failure)



One problem with this configuration, however, is there is only one point of failure; if NN1 goes down, it segments your network topology so that TDU updates will not flow. You can configure more than one common network node to provide redundancy in your network. Figure 167 is the same network, only now NN1 and NN3 are common network nodes, each with its own network segment. In this configuration, if NN1 went down, all CP-CP sessions would go down, and network connectivity would be unknown; nodes in NN3's network segment would stay up, although they would not be able to connect with any nodes on NN1's segment. Also, by linking NN1 and NN3 through the virtual routing node, the TDU updates and Locate broadcasts would be isolated to each network segment.

Figure 167 Segmented Network with Virtual Routing Node (Redundant Points of Failure)

Configuring Links to Connection Networks

To configure a link to a connection network, follow these steps:

- 1 Define the connection network to the port using:

```
ADD !<port> -APPN ConnNetworkDef [netid.]<cn name> [TG profile name]
```

This command maps the connection network to the port and, if desired, assigns a TG profile to the connection network. For example, to add a connection network named US3COMHQ.CN4 to port 4 and assign the TG profile FDDI to it, enter:

```
ADD !4 -APPN ConnNetworkDef US3COMHQ.CN4 FDDI
```

- 2 If desired, change the characteristics of the connection network using:

```
SETDefault -APPN ConnNetworkChar = <cn name> [EffectCap=<string>]
[ConnectCost=<0-255>] [ByteCost=<0-255>] [Security=<string>]
[PropDelay=<string>] [Usd1=<0-255>] [Usd2=<0-255>] [Usd3=<0-255>]
```

Using this command, you can change any or all characteristics of the connection network. For example, to change the CN4 connection network's security level to SecureCnd and byte cost to 255, enter:

```
SETDefault -APPN ConnNetworkChar = CN4 ByteCost=255 Security=SecureCnd
```

For more information on these parameters, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

You can delete a defined connection network using:

```
DELeTe !<port> -APPN ConnNetworkDef [netid.]<cn name>
```

Using Connection Networks in Boundary Routing Environments

One problem with large remote APPN networks is that if you have a lot of nodes you need to configure each remote node as an adjacent link station. Also, if you are running a Boundary Routing configuration in which a NETBuilder II bridge/router is the central site router and you have many APPN nodes at the remote site, you will have increased traffic over the WAN link every time the remote nodes initiate sessions with each other.

Figure 168 is the problem this situation can create. In this configuration, APPN nodes are on a LAN at the remote site while the network node server is the bridge/router at the central site. Because the network node server is not on the LAN, if end node A wants to initiate a session with end node B, it must first initiate an LLC2 session with the network node at the central site to discover the location of end node B. The LLC2 sessions travel over the WAN link to and from the end nodes on the LAN. If you have many nodes at the remote site LAN sending LLC2

sessions over the WAN link to the central site, this will increase traffic over the WAN link and reduce performance.

Figure 168 APPN and Boundary Routing without Remote Site Connection Network

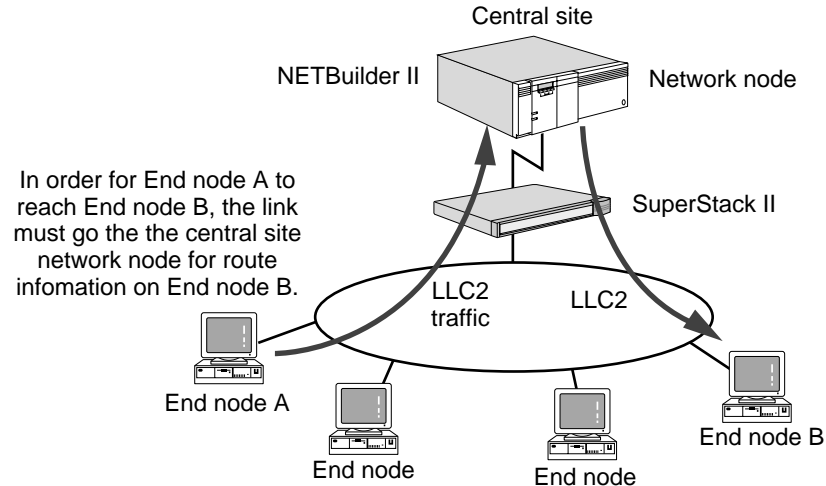
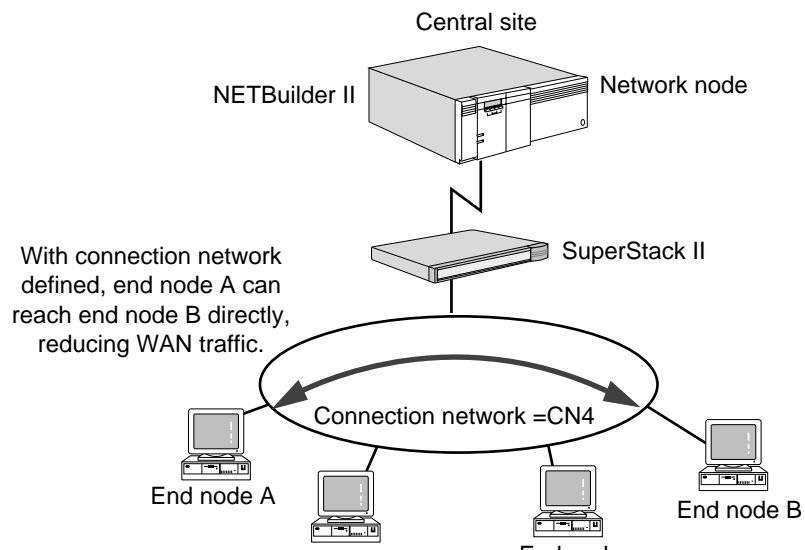


Figure 169 shows the same configuration in which a connection network has been defined for the remote site LAN. By defining all the nodes on the LAN to the connection network, the resources on that LAN are defined on the network node only once. After the resources on the LAN are defined, end node A can discover the location of end node B and initiate sessions with it directly, bypassing the central site router. This will reduce traffic over the WAN link.

You can configure more than one remote connection network. Figure 169 is an example where two different remote LANs are configured as two different connection networks from the same central site router.

Figure 169 APPN and Boundary Routing with Multiple Remote Connection Networks



Operating the Network Node

After you have configured the APPN network node and it is handling sessions properly, you can perform a number of operations to control the node and links to the node. This section describes how to do the following tasks:

- Disable the network node
- Delete adjacent link stations
- Activate and deactivate APPN ports and link stations
- Display APPN information

Disabling the Network Node

You can disable the APPN network node in one of two ways:

- Totally disable the network node and take it off the network
- Dynamically disable the network node so that when the bridge/router is rebooted, the network node is automatically re-enabled

To disable the APPN network node and take it off the network, enter one of the following commands:

```
SET -APPN CONTROL = Disable
```

or

```
SETDefault -APPN CONTROL = Disable
```

If you use the SET command, and you reboot the bridge/router, the network node will automatically be enabled. If you use the SETDefault command, you will have to re-enable the network node using the SETDefault -APPN CONTROL = Enable command if you reboot the bridge/router.

When you disable the network node, you must choose either an orderly or immediate deactivation. If you specify Immediate, the links will be deactivated first, then the ports on the network node, and then the network node itself. If you specify Orderly, the node will first be advertised as "Quiesced," the session limits will then be reset on all modes. After all ISR sessions have ended, all endpoint sessions and then all CP-CP sessions are unbound. The links are deactivated, followed by the ports on the network node, and then the network node itself. If you do not specify either, an immediate deactivation will take place.

To dynamically disable the network node with an orderly deactivation, enter:

```
SET -APPN CONTROL = Disable Orderly
```

To dynamically disable the network node with an immediate deactivation, enter:

```
SET -APPN CONTROL = Disable Immediate
```

When you disable the APPN network node, you will receive a message similar to the following:

```
Wed Dec 31 16:11:15 1993 LOCAL NETWORK NODE US3COMHQ.GOLD IS STOPPED
```

After the command is entered, the network node will not participate in the network and exchange traffic with other APPN nodes. When you disable the network node, any active sessions may be disrupted.



CAUTION: 3Com recommends that there be no active ISR sessions on the network node when you disable it. If you specify an orderly deactivation, the system will wait for all ISR sessions to go down before disabling the node.

To re-enable a previously disabled network node, enter:

```
SET -APPN CONTROL = Enable
```

Deleting Links to Adjacent Nodes

As your network needs change, you can change the network topology by deleting adjacent link stations from the network node.

To delete an adjacent link station, use the `DElete !<port> -APPN AdjLinkSta` syntax and specify the link name of the station being removed. For example, to delete the adjacent link station on port 3 with a link name of "LINK0005," enter:

```
DElete !3 -APPN AdjLinkSta LINK0005
```

To obtain a list of link names, enter:

```
SHow -APPN LinkStaCONTROL
```

Activating and Deactivating APPN Ports and Links

You can dynamically activate and deactivate APPN ports and link stations as needed. For example, if you need to deactivate a specific port for troubleshooting purposes, you can deactivate the port. You can also deactivate a specific link station on a port, also for troubleshooting purposes. By deactivating a link station you can then reactivate the link without having to redefine the link station.

Activating and Deactivating Ports

After an APPN port has been activated, if you want to change any of the configuration attributes for that port, you must first deactivate the port. After you have made your configuration changes, you then reactivate the port.

To dynamically activate or deactivate an APPN port, use:

```
SET !<port> -APPN PortCONTROL = (<Activate [NoLinkStations] |  
Deactivate [Orderly | Immediate]>)
```



This procedure applies only to ports defined for APPN using the `SETDefault !<port> -APPN PortDef` command. If the port is being used to send or receive other protocol traffic, only APPN data will be affected.

When you deactivate a port, you specify either an orderly or immediate deactivation. If you specify orderly, the system waits for all ISR sessions to terminate before deactivating the port; if you specify Immediate, the system will not wait for ISR sessions to terminate. If you specify Immediate, all sessions will first be terminated, then all LLC2 sessions will be terminated; after these processes take place, the port is deactivated. If you do not specify either, then an immediate deactivation will take place.

For example, to deactivate port 3 with an orderly deactivation, enter:

```
SET !3 -APPN PortCONTROL = Deactivate Orderly
```

After you enter the command, port 3 will be deactivated from the APPN network. If you have active link stations on that port, all links will be deactivated. When you deactivate a port, all sessions or BINDs to that port will automatically be terminated.

To activate a port and activate all the link stations on that port, enter the SET !<port> -APPN PortCONTRol command and specify "Activate." For example, to activate port 3 and activate all its defined link stations, enter:

```
SET !3 -APPN PortCONTRol = Activate
```

To activate a port but not activate any defined link stations, specify "NoLinkStations" in the command. For example, to activate port 3 but not activate any of its defined link stations, enter:

```
SET !3 -APPN PortCONTRol = Activate Nolinkstations
```

For more information on the PortCONTRol parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

Activating and Deactivating Links

After a link has been activated, if you want to change any of the configuration attributes for that link, you must first deactivate the link. After you have made your configuration changes, you then reactivate the link.

To dynamically activate or deactivate a link, use:

```
SET -APPN LinkStaCONTRol = <LinkName> <Activate | Deactivate [Orderly | Immediate]>
```

You must specify the local link station name in the command. To find out what the link name is, enter:

```
SHow -APPN LinkStaCONTRol
```

When you deactivate a link, you specify either an orderly or immediate deactivation. If you specify Orderly, the link is deactivated when all sessions are stopped. If you specify Immediate, all sessions are first stopped and then the link is deactivated. If you do not specify either, an immediate deactivation will take place.

For example, to perform an orderly deactivation for a link named "Link01," enter:

```
SET -APPN LinkStaCONTRol Link01 Deactivate Orderly
```

The link is deactivated until you enter:

```
SET -APPN LinkStaCONTRol Link01 Activate
```

For more information on the LinkStaCONTRol parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

When you activate adjacent link stations, you may receive a message on the console indicating that the CP-CP session has been activated or deactivated. For example, if you activated an adjacent link station to the node US3COMHQ.GOLD, you will receive messages similar to the following if the command was successful:

```
CONLOSER CP-CP SESSION WITH US3COMHQ.GOLD IS UP
CONWINNER CP-CP SESSION WITH US3COMHQ.GOLD IS UP
```

The first message indicates the contention loser (conloser) of the CP-CP session is up while the second message indicates the contention winner (conwinner) of the CP-CP session is up. When you deactivate adjacent link stations, you receive similar messages but they specify "DEACTIVATE."



The messages showing information on contention winners and losers only appear if the link supports CP-CP sessions, and only if CP-CP sessions exist on the link. For example, if the link is between a network node and a LEN end node, you cannot have CP-CP sessions because they are not supported on LEN end nodes.

Pinging to APPN Network Resources

Sometimes you cannot reach a given APPN network resource. Use the APpnPING command to determine if the resource is reachable without connecting to it. With APpnPING, you perform an APPC Ping to the LU in the network that you are trying to reach. To perform a ping to an LU on the network, use:

```
APpnPING [netid.]<partner_lu_name> [Mode=modename] [Size=N] [Consec=N]
  [Iterations=N] [Echo=Yes|No] [Userid=<string> [Password=<string>]]
```

For example, to ping a resource named US3COMHQ.AS400LU in batch mode with 20 iterations, enter:

```
APpnPING US3COMHQ.AS400LU Mode=#BATCH Iterations=20
```

If the APPC Ping is successful, you will receive a confirmation. If the command is not successful, you will receive a message similar to the following:

```
APPING TO US3COMHQ.GOLD DOES NOT SUCCEED
```

If you specify a userid or a password, note that these options are case-sensitive.

For more information about the APpnPING command, see the Commands chapter in *Reference for Enterprise OS Software*.

Displaying APPN Information

You can obtain different types of information regarding the APPN network, including end node and network node topology information. You can also display a list of LUs and their locations that the local network node knows about.

APPN Directory Information

The APPN directory database stores information regarding network resources and their location in the APPN network. To display a list of LU resources and their location known to the local network node, enter:

```
SHow -APPN DIRectory
```

A display similar to the following appears:

```
===== SHow -APPN DIRectory =====
-----Directory-----
Resource name      Type      Parent name      Type      Entry location   Type
US3COMHQ.CUBE     NNCP     US3COMHQ.CUBE   LOCAL    HOME             HOME
US3COMHQ.CUBE     LU       US3COMHQ.CUBE   NNCP     LOCAL           HOME
US3COMHQ.LEN1     ENCP     US3COMHQ.CUBE   NNCP     DOMAIN          HOME
US3COMHQ.LU10    LU       US3COMHQ.LEN1   ENCP     DOMAIN          HOME
US3COMHQ.NN1     NNCP     US3COMHQ.NN1    NNCP     X_DOMAIN        HOME
US3COMHQ.EN1     ENCP     US3COMHQ.NN1    NNCP     X_DOMAIN        HOME
US3COMHQ.LU7*    WILDCARD US3COMHQ.NN1    NNCP     X_DOMAIN        HOME
```

This display shows the following types of resources:

- All the local resources of the network node, which includes its own CP and LU, and all LEN nodes defined using the AdjLenDef parameter
- All adjacent end nodes and their registered resources

- All LUs in the network that the network node has discovered
- All resources defined using the DirectoryEntry command

For information on the meanings of the headings in this display, see the description of the DIRectory parameter in the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

Network Topology Information

To display a list of network nodes known by your network node, enter:

SHow -APPN NNtopology

A display similar to the following appears:

```
===== SHow -APPN NNtopology =====
-----Network Node-----
Node name           Type      RAR      Status      Function support  RSN
US3COMHQ.CN5        VRN       128      UNCONGESTED  ISR                0
US3COMHQ.CN7        VRN       128      UNCONGESTED  ISR                0
US3COMHQ.CUBE       NN        128      UNCONGESTED  ISR                2
US3COMHQ.IBM4       NN        128      UNCONGESTED  ISR                2
US3COMHQ.COM20E     NN        128      UNCONGESTED  ISR                2
```

This table may not reflect the current network node topology, which means the bridge/router network node may not be able to access all the nodes in the table. The table shows every network node the bridge/router network node has accessed historically, including nodes that may have since been removed from the network.

For information on the meanings of the headings in this display, see the description of the NNtopology parameter in the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

To display information regarding local TGs, enter:

SHow -APPN TG

A display similar to the following appears (showing one TG):

```
===== SHow -APPN TG =====
-----Network Node Transmission Group-----
Owning node name (type) = US3COMHQ.CN7          (VRN)
TG partner CP name (type) = US3COMHQ.CUBE      (NN)
TG number = 1
FRSN = 55
Days left before deletion = 15
RSN = 2
TG Status = OPERATIVE
Effective Capacity = 56000
Cost per connect time = 68
Cost per byte = 68
Security = 68
Propagation Delay = 68
User defined parameter 1 = 68
User defined parameter 2 = 68
User defined parameter 3 = 68
```

For information on the meanings of the headings in this display, see the description of the TG parameter in the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

Adjacent Link Station Information

To display a list of adjacent link stations, enter either:

```
SHoW -APPN AdjLinkSta
SHoW -APPN SdlcAdjLinkSta
```

This display shows basic information about adjacent link stations. In the display there are columns that may show the characters C, A, H, or E. These indicate support for the CPSess, AutoStart, HPR, and ErrorRecovery values, respectively. The hyphen (-) character means the value is not supported. For more information about the AdjLinkSta and SdlcAdjLinkSta parameters, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

To obtain information regarding the characteristics assigned to each adjacent link station, enter:

```
SHoW -APPN LinkStaCHar
```

Current Status of APPN Ports

To display the current status of APPN ports, use:

```
SHoW [!<port>] -APPN PortCONTRol
```

If you do not specify a port number, a display similar to the following appears:

```
===== SHoW -APPN PortCONTRol =====
-----Current Defined Ports and Status-----
Port          Port Status
!1            ACTIVE
!2            ACTIVE
!3            INACTIVE
!5            INACTIVE
```

If a port is not shown in the display, then that indicates that the port was not defined as an APPN port using the SETDefault !<port> -APPN PortDef syntax.

Active APPN Connections

To display a list of active connections, enter:

```
SHoW -APPN CONNectiOn
```

You can specify whether to display only connections to a specific node by entering the CP name of that node. You can display all connections in the network topology by entering the SHoW -APPN CONNectiOn ALL command. If you do not specify either a CP name or ALL, the display will only show connections to the local network node.

The following is a sample of the display obtained using the `SHoW -APPN CONNecTion ALL` command:

```
===== SHoW -APPN CONNecTion =====
-----Connection Topology-----
Node name                Partner name                TG num    State    RSN
US3COMHQ.CN7 (VRN)       US3COMHQ.CUBE               1         UP       2
US3COMHQ.CN5 (VRN)       US3COMHQ.IBM4               1         UP       26
US3COMHQ.CUBE            US3COMHQ.IBM4               1         UP       20
US3COMHQ.CN5 (VRN)       US3COMHQ.COM20E             1         UP       2
US3COMHQ.CUBE            US3COMHQ.COM20E             1         UP       2
US3COMHQ.IBM4            US3COMHQ.CUBE               1         UP       8
*US3COMHQ.COM20E        US3COMHQ.IBM4               1         DOWN     12
```

For more information on the `CONNecTion` parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

Current Status of Link Stations

To obtain information regarding the current status of adjacent link stations and DLUr link stations, enter:

SHoW -APPN LinkStaCONTRol

A display similar to the following appears:

```
===== SHoW -APPN LinkStaCONTRol =====
-----Current Defined Link Stations and Status-----
Port      LinkName      AdjCPName      Type      #Sess      LinkStatus
!1        @I000001     US3COMHQ.COM20E NN         4          ACTIVE
!1        LINK0000     US3COMHQ.IBM4  NN         4          ACTIVE
!1        LINK0064                    0          0          INACTIVE
```

If the entry in the `LocalLinkName` column shows the `@` character, this indicates an incoming link station that is not locally defined but was learned dynamically.

Current Status of Adjacent Nodes

To display the current status of adjacent nodes, enter:

SHoW -APPN AdjNodeStatus

A display similar to the following appears:

```
===== SHoW -APPN AdjNodeStatus =====
-----Adjacent Node Status-----
CP name      Type      TG num    Status      VRN Address      Sap      RSN
US3COMHQ.IBM4 NN         1         OPERATIVE                2
US3COMHQ.COM20E NN         1         OPERATIVE                2
```

This display shows the status of the CP-CP session between the local network node and the adjacent node. The CP name in the display is the name of the adjacent node. For more information about the data in the display, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

Intermediate Session Routing Information

Intermediate Session Routing is the intermediate routing process that takes place between the originating LU and the destination LU. Network nodes handle the Intermediate Session Routing between the originating and destination LUs. By

checking the status of ISR sessions, you can check the status of sessions routing through the node.

To obtain information regarding the current status of ISR sessions flowing through the local network node, enter:

SHoW -APPN ISRsessions

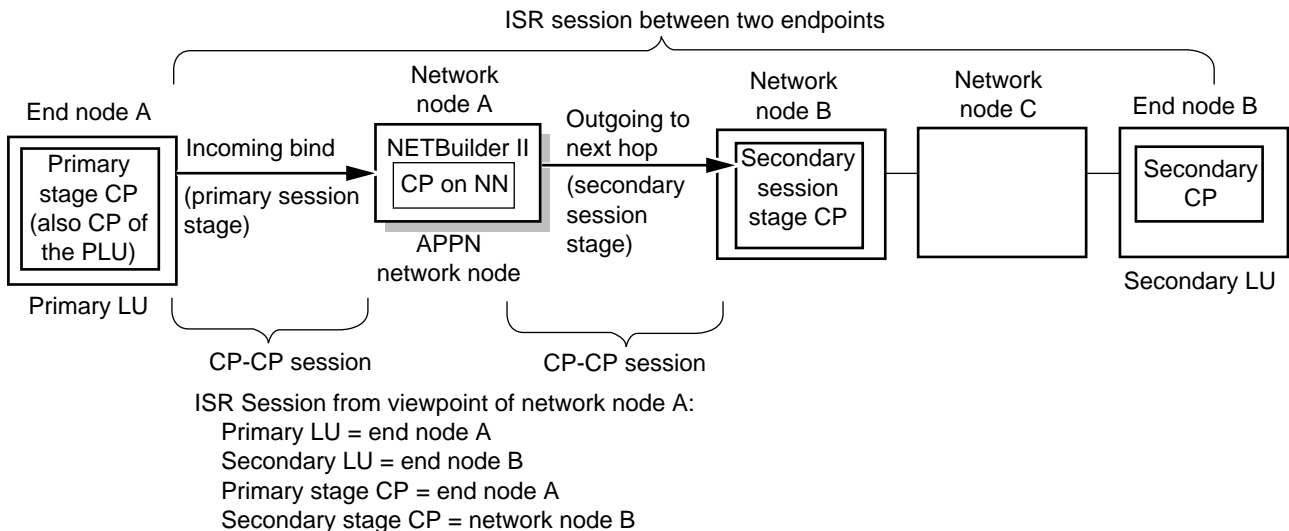
A display similar to the following appears:

```
===== SHoW -APPN ISRsessions =====
-----ISR Sessions-----
Originator CP name   COS name   Limit Res   Primary   Link name   Secondary   Link name
                   SNASVCMG   NO          LFSID     LFSID
US3COMHQ.IBM4       SNASVCMG   NO          010201   LINK0000   000201     @I000001
US3COMHQ.IBM4       #INTER    NO          010202   LINK0000   000202     @I000001
```

For more information on the headings in the ISRsessions display, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

Figure 170 is the basic concept of LU-LU sessions (routed through an intermediate node) and the relationship to CP-CP sessions. A session between two LUs spans from one endpoint LU to the other and is routed through an intermediate node. The process of routing LU-LU sessions through intermediate nodes is called *intermediate session routing*. The figure shows the relationship between the primary and secondary LUs, and the CP of the primary LU and the CP of the secondary LU, as viewed from network node A.

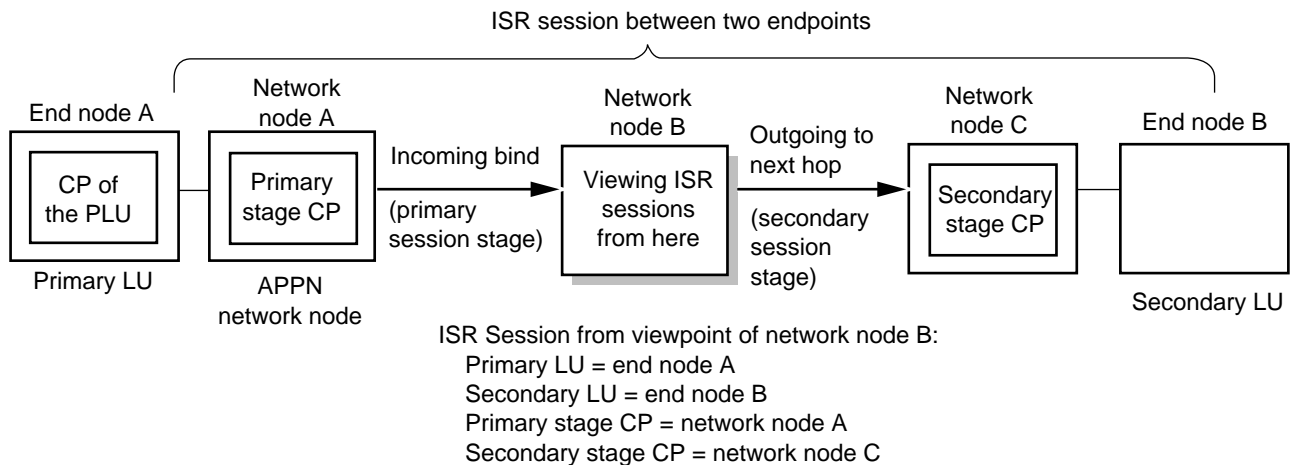
Figure 170 Intermediate Session Routing (Example 1)



The ISR session information differs depending on which network node you are viewing the session from. For example, Figure 171 is the same session example, but from the viewpoint of network node B. As shown in the figure, the primary and secondary CP information is different from the viewpoint of network node A. Also in Figure 170, end node A is the Primary Stage CP and also is the CP of the Primary LU. In Figure 171, because the network is now viewed from network node

B, network node A is the Primary Stage CP, but end node A is still the CP of the Primary LU.

Figure 171 Intermediate Session Routing (Example 2)



How APPN ISR Routing Works

APPN ISR routing works differently from other protocol routing architectures. Unlike other protocols such as IP, you do not configure static routes using APPN. Instead, network nodes maintain a directory of LU resources (and more importantly, the location of the LU resources) available in their domains. When an originating LU requests a session to a destination LU, the location of that destination LU is discovered by checking the directories on the network nodes. The actual route is determined using the APPN class of service tables. For more information on how APPN class of service tables determine the best route to take in an APPN network, see the *Configuring APPN Class of Service* chapter.

This section describes the following major topics regarding APPN concepts and how APPN network nodes facilitate APPN routing:

- APPN node types
- Role of the network node
- How the network node learns about LU resources in its domain
- How the network node learns about LUs on other adjacent network nodes, and how this information is communicated among network nodes

APPN Node Types

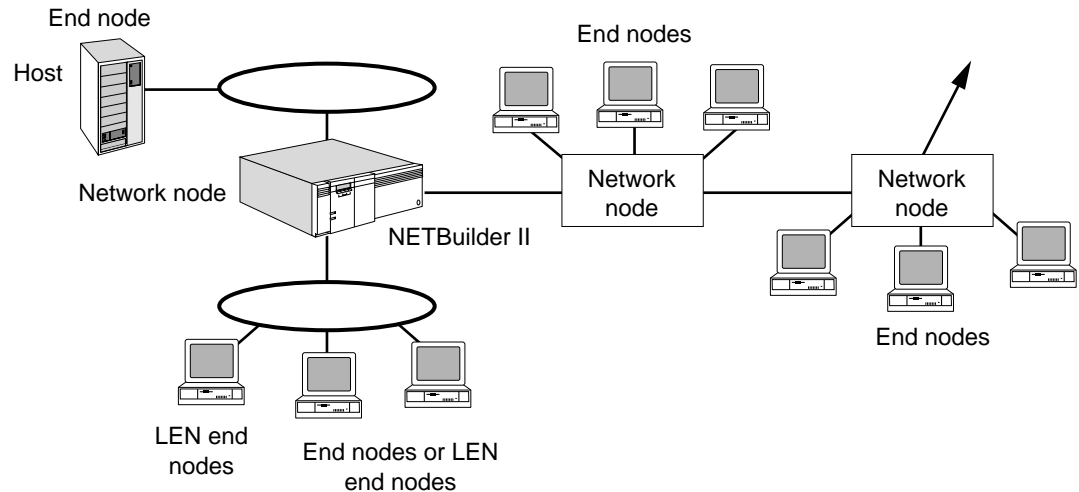
This section describes briefly the different node types defined on an APPN network. For more detailed conceptual information, see the IBM document, *APPN Architecture and Product Implementations Tutorial* and the other documents listed in "IBM APPN References" later in this chapter.

Nodes in an APPN network are divided into the following three types:

- Network nodes
- End nodes
- Low-entry networking end nodes

Figure 172 is a sample of a network topology with different devices acting as different types of nodes.

Figure 172 Node Types on an APPN Network



Network Nodes

Network nodes provide routing services and directory services for LUs on network nodes and end nodes. When a session request is initiated by a LU in the network node, an end node or LEN end node, the network node tries to locate the destination LU either on its own nodes or by querying other nodes. After the LU is located, the network node determines the best route to the destination LU according to the class of service for that session.

When a network node is added to an APPN network, the node learns network topology information from active adjacent network nodes. A network node exchanges network topology information with adjacent network nodes only when there is a change to the network topology.

When used in APPN networks, 3Com bridge/routers can function only as a network node, and does not support local LUs for application programs. Other devices that can serve as network nodes in an APPN environment include the following IBM platforms:

- IBM 6611
- S/36
- AS/400
- 3174 workstation controller (depending on the version; older versions may not be able to function as a network node)
- PCs running OS/2 Communications Manager
- IBM hosts running APPN protocols (VTAM with or without NCP supporting APPN)



This is not a complete list; other products may also be able to serve as network nodes.

End Nodes

End nodes provide limited directory and routing services for their local LUs. End nodes establish Control Point-to-Control Point (CP-CP) sessions with an adjacent network node so that LUs on the end node are available on the APPN network. The end node can also establish sessions to other LUs in the network.

The end node selects a network node to serve as its network node server and registers its LUs with the network node. By registering the end node's local resources with the network node, the network node can route any session requests from a remote node to the end node's LU. End nodes can have active connections to more than one network node at the same time, but only one network node can serve as the end node's network node server at one time.

Devices that can act as end nodes in an APPN environment include the following IBM platforms:

- AS/400
- PCs running OS/2 Communications Manager
- IBM hosts running VTAM

Low-Entry Networking End Nodes

Low-entry networking (LEN) end nodes are different from normal end nodes in that they cannot establish CP-CP sessions with a network node. As a result, LEN end nodes cannot register their resources with the network node; these resources must be predefined on the network node.

If the LEN end node has only one LU, then that LU is learned dynamically by the network node. However, if the LEN node has more than one LU, all LUs in addition to the first one must be statically defined in the network node's directory. For more information on defining LEN end node resources, see "Adding Entries" earlier in this chapter.

Many devices that are normally network nodes or end nodes can also be LEN end nodes, depending on how they are configured. Examples of devices that can be LEN end nodes in APPN networks include the following:

- IBM PCs running SAA Networking Services/2 or Networking Services/DOS (NS-DOS)
- IBM hosts running VTAM (depending on VTAM version and how it is configured)
- AS/400
- RS/6000 ANS Services/6000
- PCs running OS/2 Communications Manager

Non-IBM personal computers can also serve as LEN end nodes.

Differences Between Network Nodes and End Nodes

The primary difference between network nodes and end nodes is how each node type operates. Table 29 compares the basic differences between node types (note that LEN end nodes are a specific type of end node). For more detailed information regarding node type functionality, see the IBM document, *APPN Architecture and Product Implementations Tutorial*, in "IBM APPN References" later in this chapter.

Table 29 Functionality Differences Between Node Types

| Capability | Network Nodes | End Nodes | LEN End Nodes |
|--|---------------|-----------|---------------|
| Ability to have CP-CP sessions | Yes | Yes | No |
| Dynamically learns LU locations | Yes | No | No |
| Maintains directory of LUs and their locations | Yes | No* | No |
| Performs intermediate session routing | Yes | No | No |
| Calculates session routes | Yes | No | No |
| Supports applications via LU interface | Yes† | Yes | Yes |

* End node maintains a directory of its own LUs only.

† The NETBuilder II bridge/router provides only the routing function, and has no other application programs.

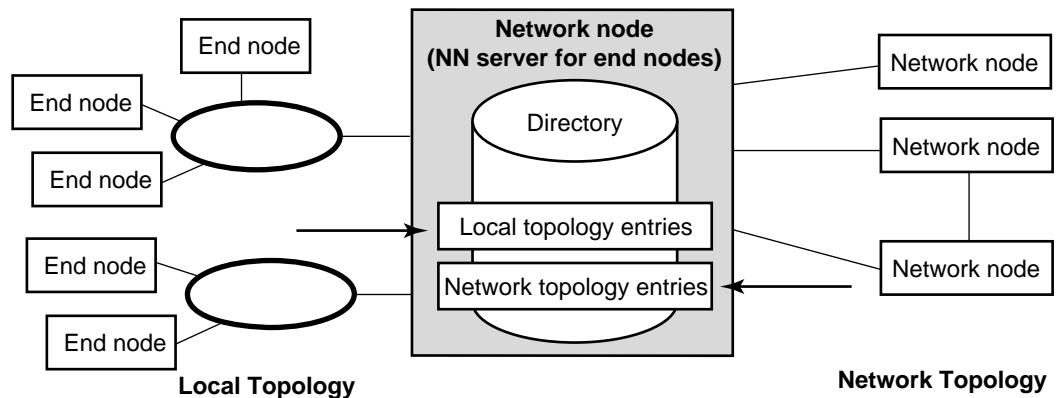
Network Node Role

The role of the network node is to provide the network services for the end nodes in its domain. It also provides the directory database that lists LUs in the local network node domain, so that the LUs can be discovered by other network nodes in the network.

The network node maintains a directory database of information for two types of nodes:

- End nodes (including LEN end nodes) in the network node's local topology
- Adjacent network nodes in the larger network topology

Figure 173 is the conceptual difference between the local topology and network topologies known by the network node.

Figure 173 Local Topology and Network Topology for the Network Node

The network node acts as the network node server for the end nodes in its domain. The network node server provides the following services to end nodes:

- Distributed directory services
 - These services locate network resources in the APPN network, and pass the information onto the end node.
- Routing services

These services calculate the best route between the origin and destination LUs based on the required class of service. For more information on how class of service calculates routes, see the Configuring APPN Class of Service chapter.



An end node can have links to more than one network node. However, only one network node can act as the end node's network node server at one time.

The network node maintains two databases:

- Directory database

These databases are LU resources on end nodes in the network node's local domain. These databases can be LUs on end nodes that were learned dynamically by the network node, or LUs on LEN end nodes that were statically defined in the directory.

- Topology database

This database maintains information regarding all network nodes and the TGs between them. The network nodes and associated TGs together make up the APPN network backbone.

How the Network Node Directory Learns About Local End Node LU Resources

APPN is a point-to-point protocol, which means that links are established between two single partner nodes. The end node maintains a direct link to the network node server.

After you configure a link from the network node to an adjacent end node, the following processes can take place:

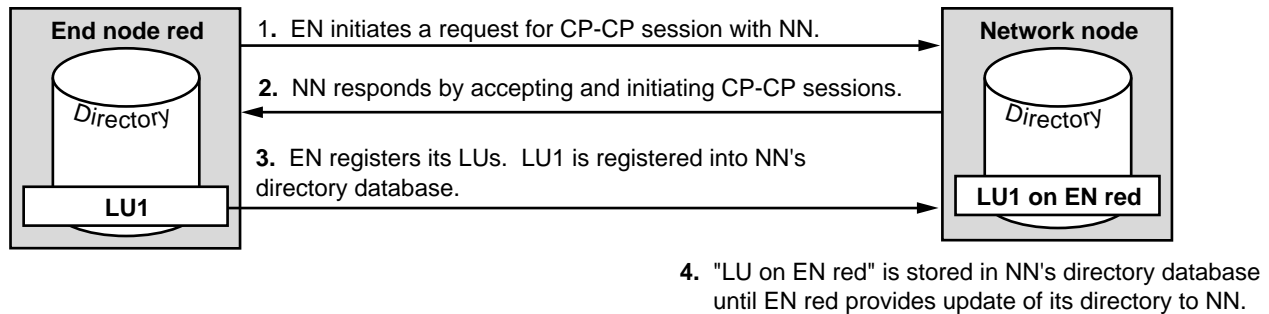
- 1 The end node calls on the network node to set up a CP-CP session (this does not apply to LEN end nodes).
- 2 The end node "registers" its LUs with the network node by sending information from the end node's local directory database to the network node.
- 3 After the network node receives the directory information, the directory entries are stored in the network node's local directory database.

These entries are temporary entries in the network node's directory database, and will change depending on how resources change on the end node. For example, if a resource on the end node is added or deleted, the information is sent to the network node's local directory database to be updated.

As long as the end node maintains a CP-CP session with the network node, the end node's resources will be registered in the network node's local database directory. After a CP-CP session is deactivated between the end node and the network node, the end node's registered entries in the network node's directory database are automatically deleted.

Figure 174 illustrates how this process works.

Figure 174 End Node Resource Registration into Network Node's Directory



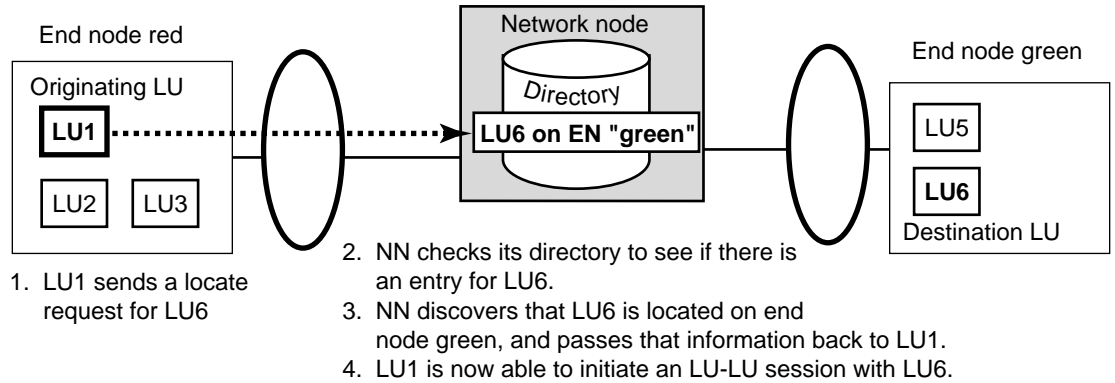
How the Network Node Discovers the Location of Destination LUs

When end nodes initiate a request for a session with a destination LU, the end node requests that the network node allocate a session to the destination LU. The network node then consults the local directory database to discover if the LU is in its domain. If the destination LU is within the network node server's domain, the network node can send the session request directly to the destination LU. However, if the destination LU is on an end node or network node, the local node may send a Locate request first. When the local node receives a positive response to the locate request, it forwards the BIND request.

Figure 175 is an example of discovering the destination in the local topology. In this example, LU1 on end node "Red" wants to initiate a session with LU6. This example assumes that CP-CP sessions are up between end node "Red" and end node "Green." Based on the examples shown in the figure, the following steps take place:

- 1 LU1, the originating LU on end node "Red," initiates a Locate request to network node A, requesting the location of LU6, the destination LU.
- 2 The network node checks the local directory database for the location of LU6.
- 3 The directory database discovers a directory entry for LU6, which shows it is located on end node "Green," and that this end node is within the network node's local domain.
- 4 The network node sends a Locate request to "Green," verifying that LU6 is still available.
- 5 "Green" sends a "locate positive" response to the network node.
- 6 The network node forwards the response to "Red."
- 7 LU1 sends a BIND to LU6 to begin process for a logical unit-to-logical unit (LU-LU) session.

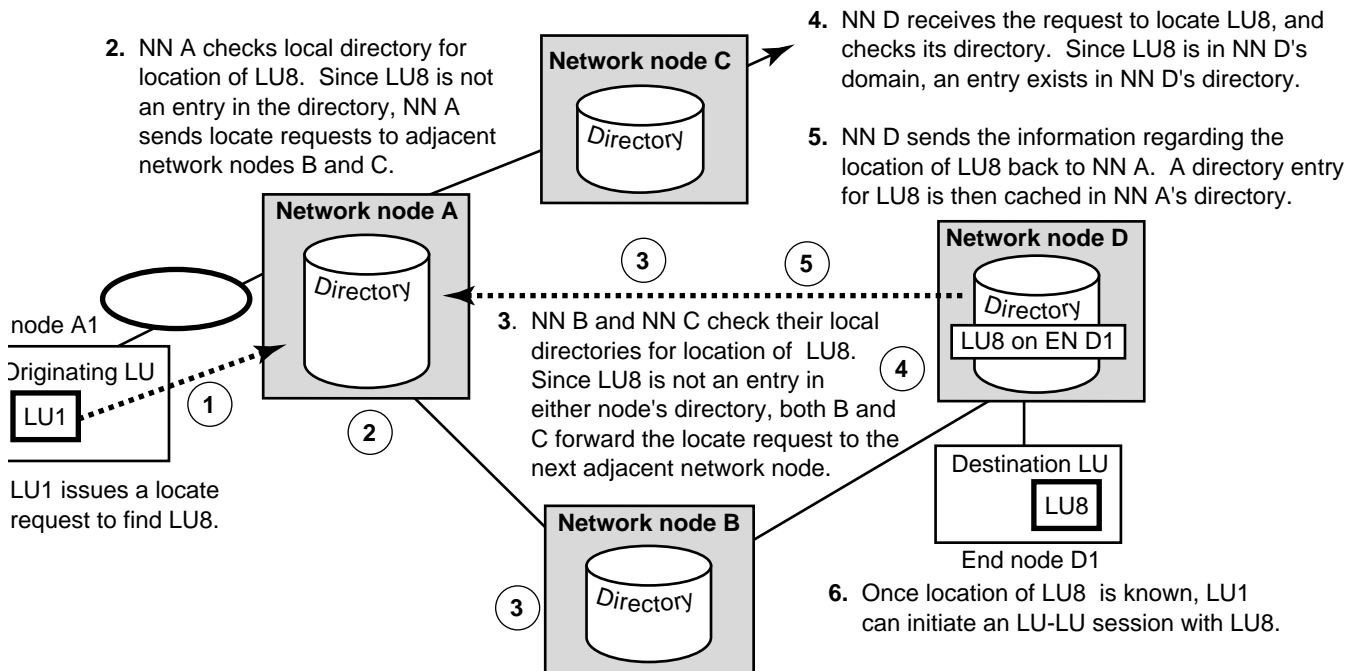
Figure 175 Discovering a Destination LU in the Local Directory



If the destination LU is not within the network node's local domain, the network node then sends locate requests to adjacent network nodes. These network nodes in turn check their directories to see if the destination LU is in their local domains.

Figure 176 is an example of discovering the destination LU in the larger network topology.

Figure 176 Discovering a Cross-Domain LU



In this example, since the destination LU is not within the local domain, the network node server sends Locate requests to adjacent network nodes. Those network nodes check their local databases for the destination LU, and if they do not find it, they in turn forward the Locate request to other adjacent network nodes. This process continues until the network node server for the destination LU finds the directory entry in its local directory database, and then forwards the information back to the first network node. The directory entry for the destination LU is then cached in the first network node's directory, in case that LU needs to be located again.

The process of locating LUs using directory services differs from the process of calculating the actual routes to those LUs. Routing is handled by topology and routing services, and routes are determined through the use of class of service tables. For more information on how class of service works, see the Configuring APPN Class of Service chapter.

Additional Information

This section provides the following additional information on some of the concepts and terminology for APPN:

- Fully qualified and not fully qualified CP name formats
- Canonical and noncanonical MAC address format options
- Setting the maximum BTU size
- APPN terminology

Fully Qualified and Not Fully Qualified CP Name Formats

When you configure the local network node, define adjacent link stations, and define adjacent nodes in the directory database, you enter the network name and CP name. However, there are different requirements and options for each.

Figure 177 shows the difference between fully qualified and not fully qualified CP name formats.

Figure 177 Fully Qualified and Not Fully Qualified CP Name Formats

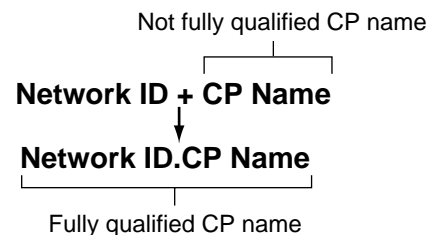
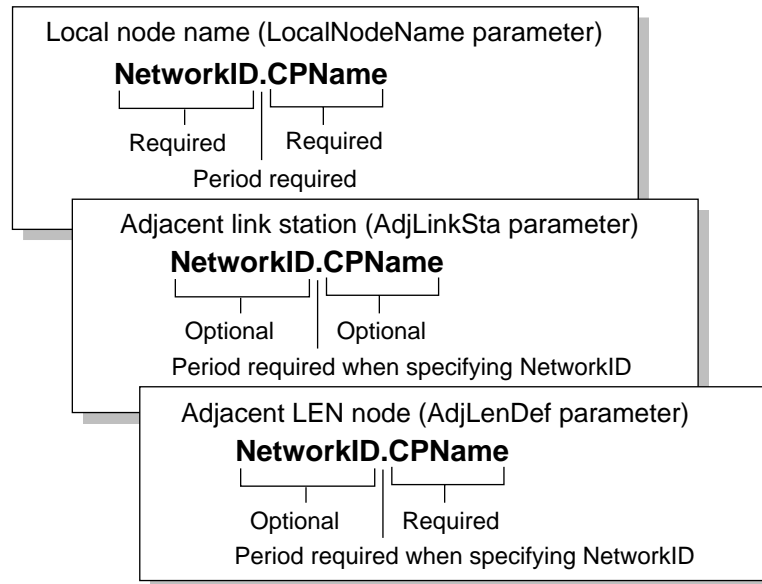


Figure 178 shows the different options for entering the CP name and network ID depending on what you are configuring.

Figure 178 Comparison of CP Name Syntax Formats

If the adjacent network ID is not present, then the system assumes that the network ID is the same as the network ID for the local network node.

MAC Address Format Options for APPN

While most SNA environments normally use noncanonical MAC address format, the default setting for the NETBuilder II bridge/router is to use canonical format in entering and displaying MAC addresses. There are two options when setting up your system with noncanonical MAC address formats:

- Change the default MAC address format by entering

```
SETDefault -SYS MacAddrFmt = Noncanonical
```

If you change the default to noncanonical, you can enter MAC addresses for APPN in noncanonical format without special notation. For more information about this parameter, see the *SYS Service Parameters* chapter in *Reference for Enterprise OS Software*.

- Change the default MAC address format to the Default setting by entering:

```
SETDefault -SYS MacAddrFmt = Default
```

If you specify Default, the system uses the appropriate MAC address for the port type. If the port type is token ring or FDDI, the system automatically displays and allows you to enter addresses in noncanonical format. All other port types would use canonical format. If you specify Default, you can still override it by preceding the MAC address with "NcMac," "Mac" or "Cmac" as described in the next paragraph.

- Precede the APPN MAC address with either "NcMac" for noncanonical or "Mac" or "Cmac" for canonical.

You have different options for using these prefixes with MAC addresses. Table 30 shows the available options. These options apply only to parameters in the APPN and SR Services.

Table 30 Options for Entering Canonical and Noncanonical MAC Addresses*

| Canonical format where C=canonical† | Noncanonical format where N=noncanonical† |
|-------------------------------------|---|
| Cmac_XXXXXXXXXXXX | Ncmac_XXXXXXXXXXXX |
| Cmac_%XXXXXXXXXXXX | Ncmac_%XXXXXXXXXXXX |
| Cmac%XXXXXXXXXXXX | Ncmac%XXXXXXXXXXXX |
| C%XXXXXXXXXXXX | N%XXXXXXXXXXXX |
| C_%XXXXXXXXXXXX | N_%XXXXXXXXXXXX |
| C_XXXXXXXXXXXX | N_XXXXXXXXXXXX |
| CXXXXXXXXXXXX | NXXXXXXXXXXXX |

* If you do not set the -SYS MacAddrFmt parameter.

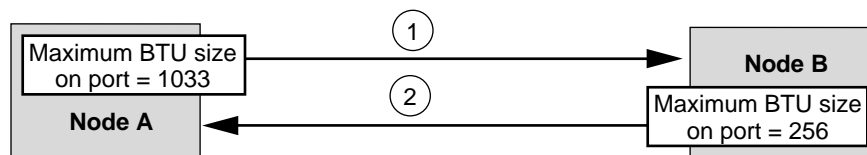
† Underscores indicate spaces.

Setting the Maximum BTU Size

When you configure adjacent link stations, one of the values you need to set is the maximum basic transmission unit (BTU) size. This value determines the maximum BTU size that will be allowed over the link.

When you configure partner nodes, each port on both sides of the link may have different maximum BTU sizes. During link station negotiation, each node communicates the maximum BTU size value it accepts. The lower of the two maximums is used to prevent the port with the lower capacity from over using its available memory capacity.

Figure 179 shows the process of how BTU size negotiation takes place.

Figure 179 BTU Size Negotiation

- ① As part of session negotiation, Node A tells Node B the maximum BTU size allowed on port is 1033.
- ② In response, Node B informs Node A that its maximum port BTU size is 256.

As a result of negotiation, the session uses the smaller BTU size of 256 in both directions. The smaller Btu size is always used to prevent a port from receiving larger BTUs than it can handle.

The maximum BTU size will differ depending on what physical medium is being used over the port. The recommended maximum BTU size allowed over APPN ports is 2,057, which equals the maximum ISR RU size of 4,096 + 9. (Certain media may allow larger frame sizes, but for the best buffer use on APPN ports, the BTU size should not be larger than 5,005.)

If the physical port medium is Ethernet, the recommended maximum BTU size is 1,500. If you are using serial lines for bridging LLC2, the recommended maximum BTU size for Source Route Transparent bridging is 1,500, while the recommended maximum for Source Route bridging is 5,005.

APPN Terms

A list of important terms that are used in this chapter is provided here to briefly explain APPN routing concepts.

| | |
|--|---|
| adjacent link station | The local information regarding a link to an adjacent node. It is the link definition stored in the network node. |
| adjacent node | A node immediately adjacent to the local network node. You define adjacent nodes in the network node's directory. |
| connection network | A configuration in which a set of APPN nodes are grouped together with one logical name to help reduce the number of direct links required and the amount of broadcast traffic. |
| control point (CP) | An entity that manages T2.1 nodes and their resources. In APPN, the control point initiates links to adjacent nodes, and exchanges CP capabilities with adjacent nodes when CP-CP sessions are established. |
| control point-to-control point (CP-CP) session | Takes place between two adjacent nodes, to exchange routing and resource information, as well as the CP capabilities of the node. CP-CP sessions can take place between two network nodes, between a network node and an adjacent end node, and between two end nodes. (Note that LEN end nodes do not support CP-CP sessions.) Not all links can support CP-CP sessions. |
| dependent LU requester (DLUr) | Assists PU type 2.0 and 2.1 nodes with dependent LUs that require the services of a remote SSCP. The DLUr obtains these SSCP services from the dependent LU server (DLUs) and in turn provides the services to the dependent LUs. In the 3Com APPN implementation, the NETBuilder II bridge/router acts as the DLUr. |
| dependent LU server (DLUs) | A host that provides SSCP services to a dependent LU requester (DLUr). |
| directory | Resides on the network node and provides a list of logical units (LUs) on the local and network topologies, and the locations of those LUs. (Note that an end node also has a directory, but it only lists the end node's local LUs.) |
| end node | A node with LU resources that can initiate LU-LU sessions. A regular end node (as opposed to a LEN end node) can have CP-CP sessions with the network node acting as the end node's network node server. End nodes do not support intermediate session routing. |
| Intermediate Session Routing (ISR) | The routing that takes place through intermediate nodes between the originating LU and the destination LU. The NETBuilder II bridge/router acting as the network node provides intermediate session routing. |
| low-entry-networking (LEN) end node | A Type 2.1 node that does not have a control point. Without the control point, the LEN end node does not have the ability to hold CP-CP sessions with a network node. As a result, the network node cannot learn the LEN end node's LUs dynamically. The LEN end node's LUs must be statically defined in the network node's directory. |

| | |
|----------------------|--|
| logical unit (LU) | Provides an interface for applications to communicate and gain access to an SNA network. The network node learns LUs on adjacent network nodes and end nodes dynamically, while LUs on LEN end nodes must be statically defined in the directory. |
| network node | The backbone of the APPN routing architecture. Network nodes provide intermediate session routing between two end stations, exchange directory and topology information with adjacent network nodes, and provide routing services for end nodes in their domain. |
| network node server | The network node that “serves” the end nodes in its domain by maintaining a list of LUs on the end nodes, so that incoming LU requests can find the location of a destination LU. The network node calculates routes for LUs on the end nodes. |
| partner node | Two adjacent nodes that have configured each other as adjacent link stations so they can have links with each other. (Partner node is not an IBM APPN term, and is used here for conceptual purposes only. The term is not to be confused with the IBM terminology for partner LUs.) |
| transmission group | A link between two nodes. |
| virtual routing node | A logical representation of a defined connection network between two nodes. |

IBM APPN References

The following IBM documents provide additional information for IBM's APPN and SNA implementation:

APPN Architecture and Product Implementations Tutorial, International Business Machines Corporation, April 1994 (GG24-3669-02)

IBM Systems Network Architecture: LU6.2 Reference: Peer Protocols, International Business Machines Corporation (SC31-6808)

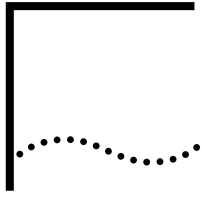
IBM Systems Network Architecture: APPN Architecture Reference, International Business Machines (SC30-3422)

IBM Systems Network Architecture: Management Services Reference, International Business Machines (SC30-3346)

IBM Systems Network Architecture: Formats, International Business Machines (GA27-3136)

IBM Systems Network Architecture Concepts and Products, International Business Machines (GC30-3072)

IBM Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2, International Business Machines (SC30-3269)



APPN HIGH PERFORMANCE ROUTING

This chapter describes how to configure your 3Com bridge/router to perform Advanced Peer-to-Peer Networking (APPN) High Performance Routing (HPR).

High Performance Routing is an advanced method of routing APPN sessions that provides greater scalability and performance than Intermediate Session Routing (ISR), the original APPN routing method. The improvements that HPR provides over ISR include:

- Dynamic rerouting if a link on a path fails, which enables the connection to stay up.
- Streamlined routing at the Systems Network Architecture (SNA) Path Control layer.
- An architecture designed to take advantage of high-speed media.
- Intermediate nodes do not have to process message segmentation, which reduces overhead.
- Intermediate nodes do not buffer messages, which reduces overhead.

You can configure ports on the NETBuilder II bridge/router to perform either ISR or HPR. By default, HPR is enabled on APPN ports and adjacent link stations. If you want to configure specific ports or link stations for ISR, you must disable HPR on those ports or link stations. For more information about configuring the bridge/router as an APPN network node for ISR, see the Configuring APPN Intermediate Session Routing chapter.



For conceptual information about HPR, see “How HPR Works” later in this chapter.



APPN routing is supported only on NETBuilder II bridge/routers with DPE modules.

Configuring the Network Node to Perform HPR

HPR networks operate over network nodes and end nodes like ISR networks. The NETBuilder II bridge/router can be configured as a network node only; because the bridge/router does not provide any application programs on the SNA network, it cannot act as an end node or LEN end node.

When you configure the NETBuilder II bridge/router as an HPR network node, it can function as a Rapid Transport Protocol (RTP) tower node. For an explanation of RTP tower nodes and the other types of HPR network nodes, see “HPR Node Types” later in this chapter.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the procedures described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

- If necessary, use LAN Address Administration (LAA) to reassign media access control (MAC) addresses for paths that will send and receive APPN traffic. You must perform this configuration *before* starting APPN. For more information on configuring LAA, see the Configuring LAN Address Administration chapter.
- If you are planning to support both APPN and DECnet on the same bridge/router, you must configure DECnet *before* configuring APPN. Configuring DECnet can change MAC addresses, which would affect any existing APPN configuration. For more information on configuring DECnet, see the Configuring DECnet Routing chapter.
- If necessary, configure the Logical Link Control, type 2 (LLC2) data link interface or the Data Link Switching (DLSw) interface for the ports you will use for APPN traffic. For more information on configuring the LLC2 data link interface, see the Configuring the LLC2 Data Link Interface chapter. For more information on configuring DLSw, see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.
- If you will be sending APPN traffic over synchronous data link control (SDLC) lines, configure the bridge/router for SDLC operation first. For more information on SDLC configuration, see the Configuring Synchronous Data Link Control Connectivity chapter.
- If you will be sending APPN traffic over Frame Relay, configure the Frame Relay interface before configuring the APPN network node. For more information on configuring Frame Relay, see the Configuring Wide Area Networking Using Frame Relay chapter.
- Configure the NETBuilder II bridge/router as an APPN network node following the procedures in the Configuring APPN Intermediate Session Routing chapter. Using those procedures, you first set up the basic framework for your APPN configuration using ISR. To bring the APPN network node to the HPR level, follow the procedures in this chapter.

Procedure To set up the bridge/router network node to perform HPR, follow these steps:

- 1 Set your APPN ports to support HPR by performing one of the following steps:
 - a If you did not disable HPR when first setting up the ports as described in the Configuring APPN Intermediate Session Routing chapter, you need not change anything. The port will already support HPR.
 - b If you are converting a port from ISR mode to HPR, set the APPN port to perform HPR using:

```
SETDefault !<port> -APPN PortDef = <DLC type>
(LLC2|FR|PPP|DLSW|SDLC|UNdef) <max_btu_size>(99-8192)
[HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

Make sure to specify HPR=Yes, and specify whether you want the port to provide link level error recovery. You can specify the other optional values of the PortDef parameter as desired.

Setting ErrorRecovery to Yes provides link-level error recovery for the incoming connection only. If you want link-level error recovery for the outgoing connection, then you must specify Yes for the ErrorRecovery value when setting the AdjLinkSta or SdlcAdjLinkSta parameter. If you set the port data link control (DLC) type to DLSW or SDLC, then link-level error recovery is enabled by default. If you set the port DLC type to LLC2, FR, or PPP, then you must specify error recovery support if desired.

If you use error recovery, it will create additional overhead on the link.

To configure port 7 for Frame Relay at a maximum basic transmission unit (BTU) size of 1033 and to enable support for HPR and error recovery, enter:

```
SETDefault !7 -APPN PortDef = FR 1033 HPR=Yes ErrorRecovery=Yes
```

- 2 Define adjacent link stations for HPR by performing one of the following steps:
 - a If you did not disable HPR when first setting up the adjacent link station as described in the Configuring APPN Intermediate Session Routing chapter, you need not change anything. As defined on the bridge/router, the link station will already support HPR.

- b Define each adjacent link station to support HPR using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn)
  <max_btu_size>(99-8912) [[Cmac|Ncmac] dest media addr] [Sap=<num>]
  [CPName=[netid.]cpname] [Nodeid=<ID>] [LinkName=<name>]
  [TGprof=<name>] [AutoStart=(Yes|No)] [CPSess=(Yes|No)]
  [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

Make sure to specify the adjacent link station as an HPR node by specifying HPR=Yes. If you want link level error recovery for the outgoing connection, specify ErrorRecovery=Yes. You can specify the other optional values as desired.

- c If using SDLC, define each adjacent link station to serve as an HPR node using:

```
ADD !<port> -APPN SdlcAdjLinkSta <type>(NN|EN|Learn)
  <max_btu_size>(99-8912) <station addr>(Hex 1-FE)
  [CPName=<[netid.]cpname] [Nodeid=<ID>] [LinkName=<name>]
  [TGprof=<name>] [AutoStart=(Yes|No)] [CPSess=(Yes|No)]
  [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)] [SendWindow=<num>]
  [ContactTimer=<num>] [NoRspTimer=<num>]
  [NoRspTimRetry=<num>]
```

Make sure to specify the adjacent link station as an HPR node by specifying HPR=Yes. If you want link level error recovery for the outgoing connection, specify ErrorRecovery=Yes. You can specify the other optional values as desired.

When you configure HPR over SDLC connections, HPR must be enabled on both sides of the SDLC connection.

For more information on the full syntax of these parameters, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.



APPN over SDLC connections is supported on the NETBuilder II HSS-3-Port V.35 module only.

- 3 If you have not done so already, define the link characteristics using:

```
SETDefault -APPN LinkStaChar = <LinkStation name> [EffectCap=<string>]
  [ConnectCost=<0-255>] [ByteCost=<0-255>] [Security=<string>]
  [PropDelay=<string>] [Usd1=<0-255>] [Usd2=<0-255>] [Usd3=<0-255>]
```

- 4 To enable the bridge/router to function as an APPN network node, enter:

```
SETDefault -APPN CONTrol = Enable
```

After HPR has been configured on the network node and the node has been enabled, the network node can participate in the HPR network. If the network node is part of an ISR environment, an HPR subnet can be created. Depending on how you set up your network, the network node can be either an HPR endpoint (for Rapid Transport Protocol (RTP) connections) or an HPR intermediate node (for Automatic Network Routing). An HPR node does not become an RTP endpoint until it accepts a session through it.



Not all devices that support APPN support HPR. If you configure an adjacent link station to a device that does not support HPR, RTP connections cannot take place, but ISR sessions can.

Configuring HPR Subnets within ISR Networks

After you have configured two adjacent network nodes for HPR, and they have established the appropriate RTP connection, an HPR subnet is created, even if the two nodes reside within an APPN ISR network. When you create a mixed HPR and ISR environment, you only gain the benefits HPR provides on those links where both partner nodes support HPR. On links where one node is HPR-capable and the partner node is not, the link defaults to normal APPN ISR operation.

Figure 180 is an example in which two network nodes within an APPN ISR network have been upgraded to support HPR. In this example, only the links between the HPR-capable nodes support HPR operation, including RTP connections. Since there are alternate paths that are ISR only, you will not gain the benefits that HPR provides because the topology routing services and class of service used to calculate routes do not provide greater weight to the HPR paths over ISR paths. As a result, mixing HPR nodes and ISR nodes in this type of configuration is not recommended.



*You can create customized class of service tables to prioritize HPR paths over ISR paths, but you will need to do extra configuration. For more information on customizing class of service, see the *Configuring APPN Class of Service* chapter.*

Figure 180 HPR Subnet within APPN ISR Network

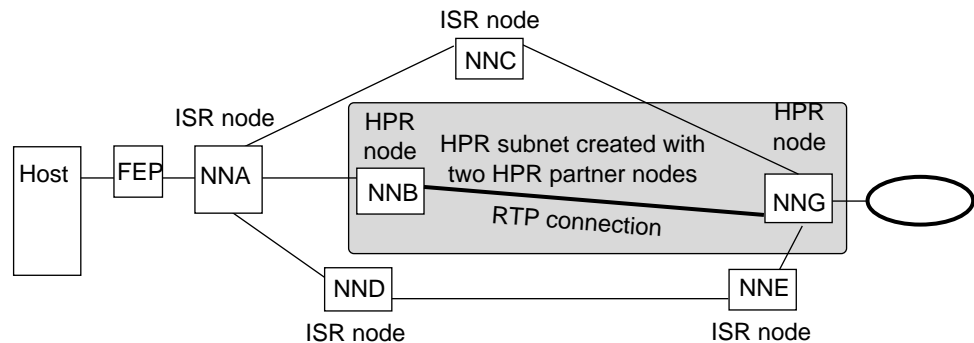
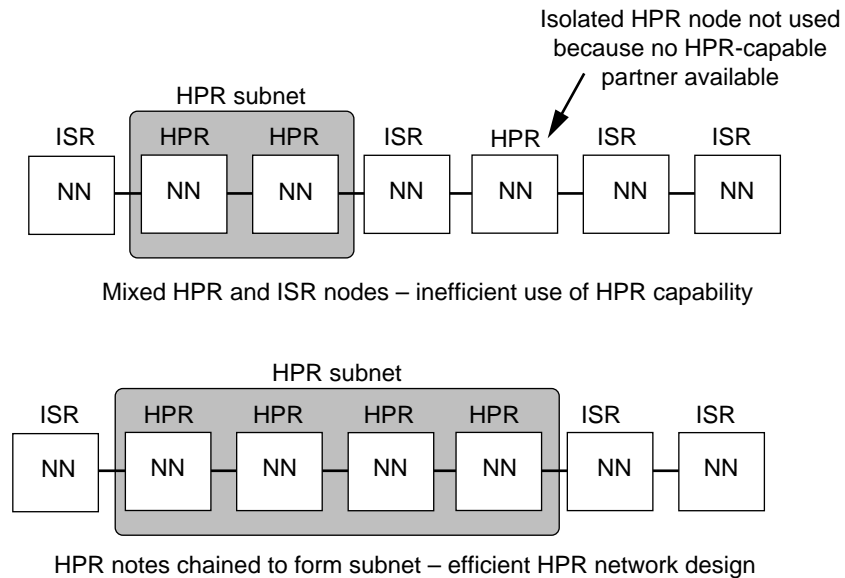


Figure 181 is an example in which HPR nodes and ISR nodes are mixed over a path. In the top example, two HPR nodes form an HPR subnet, and an ISR node is between two HPR nodes. The result is that the third HPR node is isolated, and the benefits of HPR are limited to only the HPR subnet. In the bottom example, the ISR node is converted to support HPR, which chains all four HPR nodes together to

form a larger HPR subnet, extending the benefits of HPR over a larger portion of the network.

Figure 181 Chaining HPR Nodes to Form HPR Subnet

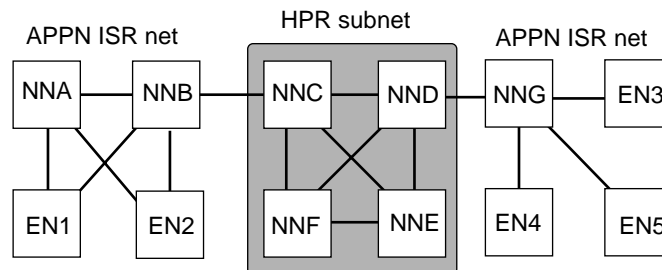


3Com recommends that you design your network so that HPR nodes reside either:

- On the network backbone between APPN ISR segments.
- Within a self-contained HPR subnet in which there are no alternate ISR paths.

By configuring HPR nodes within a larger HPR subnet on the network backbone, you gain the high-speed benefits and processing efficiencies that HPR provides. Figure 182 is an example of an HPR subnet used on a network backbone connecting multiple ISR networks.

Figure 182 HPR Subnet on a Network Backbone

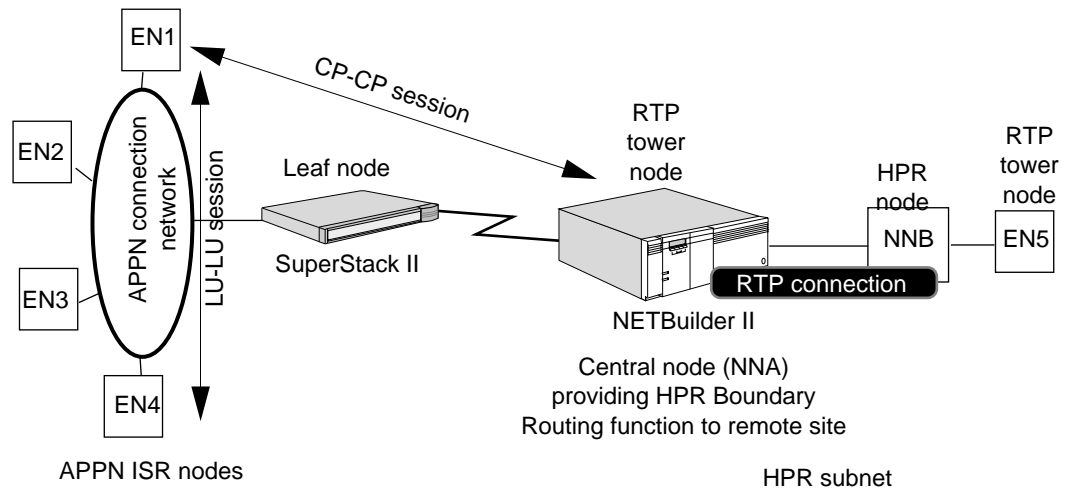


Using HPR with Boundary Routing Environments

You can configure HPR with Boundary Routing so that the NETBuilder II bridge/router acting as an HPR node is also acting as the central node in the Boundary Routing topology. Because the Superstack II NETBuilder bridge/router, acting as the leaf node at the remote site, does not support APPN in either ISR or HPR mode, the NETBuilder II bridge/router at the central site provides the HPR boundary function (to translate ISR traffic to HPR and vice-versa).

Figure 183 is an example in which HPR is configured with Boundary Routing at a remote site where an APPN connection network has also been configured. In the example, network node A (the central node) is an RTP tower node for HPR and is maintaining RTP connections with network node B. The central node is also providing the boundary function to the APPN end nodes at the remote site. For more information about Boundary Routing system architecture, see the Configuring Boundary Routing System Architecture chapter.

Figure 183 HPR and Boundary Routing



Operating the HPR Network Node

After the network node has been configured for HPR, you can perform tasks such as setting RTP connection timers and initiating nondisruptive path switching.

Setting RTP Connection Timers

To set the timers for RTP connections, use:

```
SetDefault -APPN HprTimer = [AliveTimer=<30-600>]
[PathSwitchTimerLow=<240-960>][PathSwitchTimerMed=<120-480>]
[PathSwitchTimerHigh=<60-240>][PathSwitchTimerNtwk=<30-120>]
```

Using this command, you set the timer settings for the RTP connection. Changing this parameter only affects new RTP connections, and has no effect on existing RTP connections. The options allow you to set the path switch timer for connections with low, medium, high, or network priority.

Displaying RTP Connections

To display a list of RTP connections, use:

```
SHow -APPN RTP [name]
```

To display statistical information regarding RTP connections, use:

```
SHow -APPN RTPStats [name]
```

For information about the contents of these displays, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

Initiating a Nondisruptive Path Switch

A nondisruptive path switch can be triggered for several reasons, such as a local or remote link failure, or an RTP connection failure detection. When one of these failures occurs, the system initiates the path switch to try to determine if an

alternate path is available. Normally, this nondisruptive path switching occurs automatically.

Using the PathSwitch command, you can request that the system switch an RTP connection to an alternate path. When a path switch is initiated, the system checks all available paths in the HPR topology to determine if a more desirable path is available. If a more desirable path is available, the RTP connection switches to that path; if the current path is the most desirable, then the system remains at the current path.

To initiate a nondisruptive path switch, enter:

```
PathSwitch <RTP name>
```

You must specify the RTP connection name that you want the system to switch.

To obtain a list of RTP connection names, enter:

```
SHow -APPN RTP
```

You cannot specify the new path to switch to; the system determines which path to switch to.

You can only switch paths from one HPR path to another; you cannot switch an RTP connection to a path running APPN ISR traffic.

For more information about nondisruptive path switching, see "Nondisruptive Path Switching" later in this chapter.

How HPR Works

High Performance Routing is designed to work in conjunction with APPN Intermediate Session Routing (ISR) network nodes. HPR nodes perform many of the same functions as ISR nodes. For example, HPR nodes use the same method of calculating routes based on the Topology Routing Service database and class of service tables. HPR nodes also supports such APPN features as connection networks and support for parallel transmission groups (TGs).

In the HPR architecture, both partner nodes must support HPR for RTP connections to take place between the nodes. If one node supports HPR and the partner node does not, then the link will support ISR functionality only.

For more complete information regarding HPR, see the IBM document *APPN Architecture and Product Implementations Tutorial* (GG24-3669-92).

HPR Node Types

There are two different levels of HPR node functionality:

- Base HPR node

Base HPR nodes support Automatic Network Routing (ANR) and can only act as intermediate nodes in an RTP connection. Base HPR nodes cannot be the endpoint of an RTP connection. The 3Com bridge/router cannot act as a base HPR node.

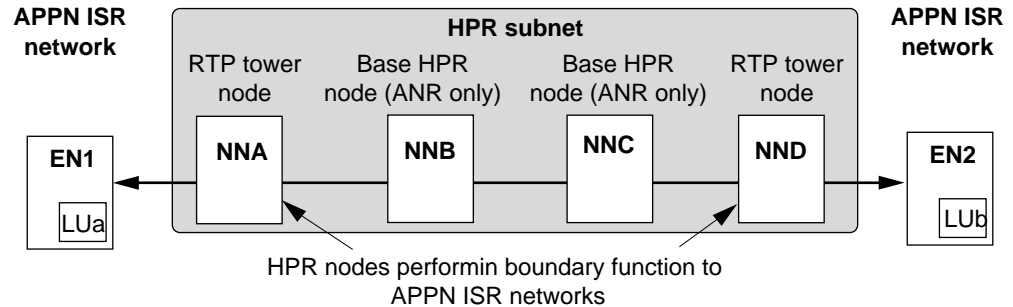
- RTP Tower node

RTP tower nodes can be either RTP endpoints or RTP connection intermediate nodes performing the ANR function. When acting as RTP endpoints, RTP tower nodes perform adaptive-rate-based flow control. If the RTP tower node is connected to an APPN ISR subset, then it performs the boundary function,

which joins a session in the HPR subnet with a session in the APPN ISR subnet. The 3Com bridge/router network node acts as an RTP tower node in the HPR network.

Figure 184 shows the relationship of the different node types in an HPR network.

Figure 184 HPR Node Types



IBM Devices Supporting HPR

For the APPN HPR network node to provide HPR functionality on a link, the partner node device must also support HPR. IBM devices that support HPR include:

- VTAM V4R3/NCP V7R3
- OS/400 V3R1 (ANR only)

This list is not complete, and other devices may support HPR in the future. HPR can be supported on APPN network nodes and end nodes.

Automatic Network Routing

Automatic Network Routing is a source routing protocol used to route LU6.2 session and control traffic from node-to-node through an HPR network or subnet. ANR operates at the lower end of the SNA Path Control layer.

Unlike most SNA traffic, which is normally connection-oriented, ANR packets are connectionless, and HPR routes these network layer packets independently. These packets contain a network layer header that carries routing information. Because the routing information is processed at the network layer, this processing is more efficient than the processing for ISR packets.

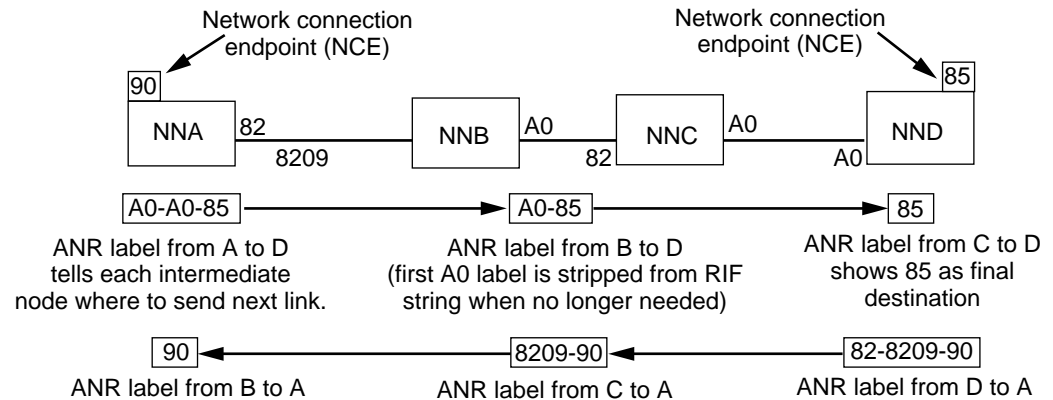
The routing information is contained in the ANR routing field, which consists of a string of ANR labels. Each label describes the path from one node to the next immediate node; the ANR label string describes the path from the source HPR node to the destination HPR node of the RTP connection.

When an HPR node receives an ANR packet, it checks the first label of the ANR routing field and uses that label to determine which link to send the outgoing packet over. That label is then stripped from the ANR routing field, so the receiving node can check the next label in the ANR routing string.

Figure 185 shows how ANR routes network layer packets and how ANR labels are used to route the packets from node-to-node, and then are stripped when they are no longer needed. In the figure, the ANR label from network node A to

network node D is A0-A0-85, and at each intermediate node the first part of the label is stripped from the packet.

Figure 185 ANR Label Processing



The ANR label is from 1 to 8 bytes long and is of local significance on the node only. ANR labels only need to be unique on the local node, not on the larger network. In the figure, the label A0 is used several places, but the duplication is acceptable as long as the A0 label is unique on each node. In addition, ANR labels can be of different sizes within a node.

Rapid Transport Protocol

Rapid Transport Protocol is a reliable connection-oriented protocol that HPR uses to carry session traffic through an HPR network. It routes logical unit-to-logical unit (LU-LU) session traffic flows between the two RTP connection endpoints using the ANR routing method. RTP provides the following features:

- Full duplex transmission and delivery of messages in sequence
- Message segmentation and reassembly
- Selective retransmission, in which only the portions of data that are lost are retransmitted
- Adaptive-rate-based congestion and flow control

RTP Connections

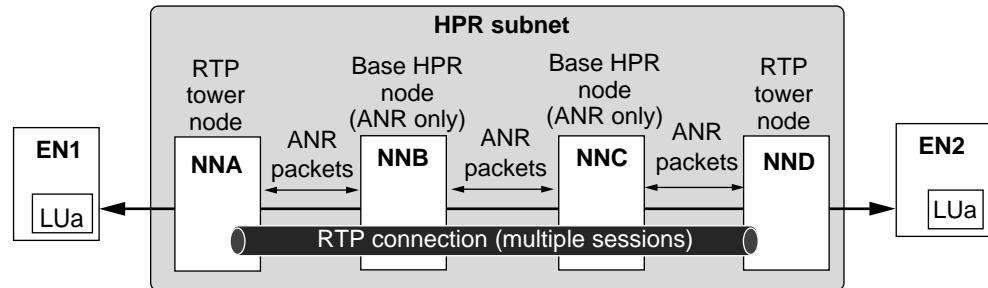
RTP connections are logical connections between two nodes over a specific path in an HPR network. These logical connections are used to transport full-duplex session traffic end-to-end between the two nodes. RTP connections support a single class of service on each connection, enabling all traffic on the connection to use the same transmission priority. You can multiplex multiple sessions of the same class of service over one RTP connection, but all traffic on a given session must flow on the same RTP connection. If you have multiple sessions with different classes of service, then the bridge/router uses different RTP connections for each class of service.

Figure 186 is an example of an RTP connection across several nodes in an HPR subnet. LUa on EN1 is connected to LUb on EN2. In between the two end nodes is an HPR subnet, with the RTP connection spanning across it.

You can have multiple sessions with the same class of service on an RTP connection. All traffic using a specific class of service travels on the same RTP connection (the 3Com HPR implementation does not support sending traffic of

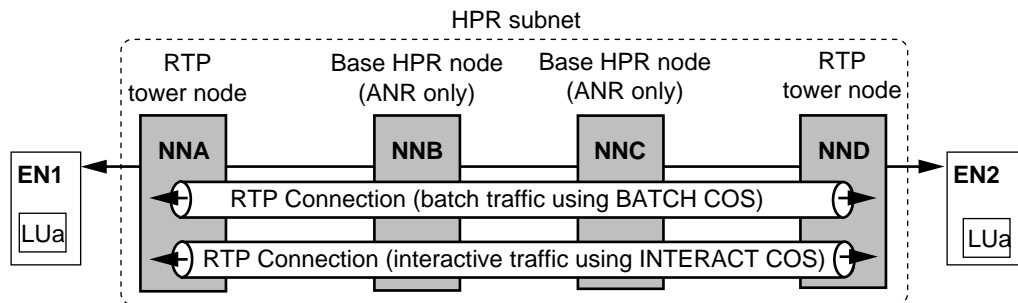
the same class of service over different RTP connections). The figure also shows the ANR routing packets being forwarded from node-to-node.

Figure 186 RTP Connection Across HPR Subnet



You can have multiple RTP connections between HPR nodes, with each RTP connection handling a different class of service. In Figure 187, there are multiple RTP connections. One RTP connection is used for batch sessions (using the BATCH class of service), and one RTP connection is used for interactive sessions (using the INTERACTIVE class of service).

Figure 187 Multiple RTP Connections Using Different Classes of Service



Nondisruptive Path Switching

Through RTP, HPR provides nondisruptive path switching, which enables the node to switch an RTP connection to a new path if the current path fails or the link fails. When the system initiates a path switch, it attempts to switch the RTP connection to the most desirable path at the time. This process enables dynamic rerouting in case of link failure, and the rerouting takes place fast enough not to disrupt the active sessions. The most desirable path at the time is the HPR-only route with the lowest weight. Even if there is an alternative ISR-only path that has a lower weight than the lowest-weight HPR route, the lowest weight HPR route is chosen.

The node can trigger a path switch in one of the following situations:

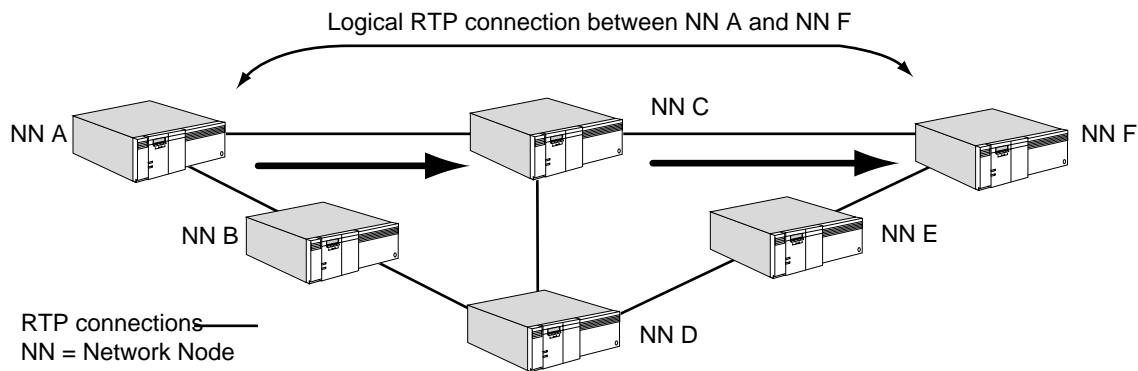
- RTP connection failure detection
 - When an RTP endpoint periodically sends out a status request to its partner, if a reply is not received within the specified time set by the `HprTimer` parameter, the RTP endpoint sends a state exchange request to determine the status. If this state exchange fails after several retries, then the RTP endpoint determines that the RTP connection failed and triggers a path switch.
- Local link failure

If a local link associated with an RTP connection fails, the system can initiate a path switch faster than relying on the RTP connection failure detection.

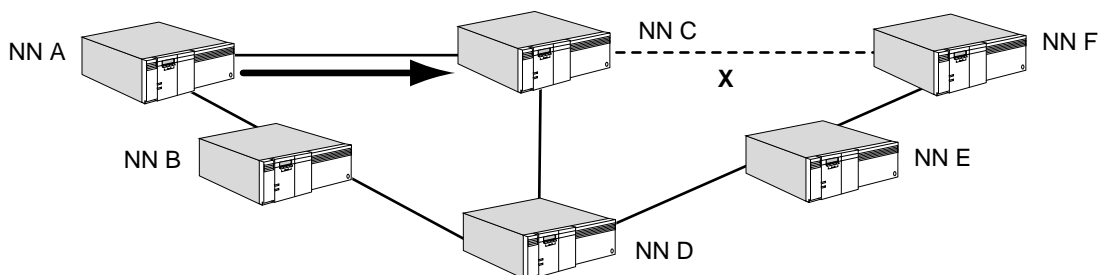
- Initiated by the user using the PathSwitch command.

When a user initiates a path switch, the system checks to determine the most desirable path to switch the RTP connection to. If the current path is the most desirable path, the system remains at the current path.

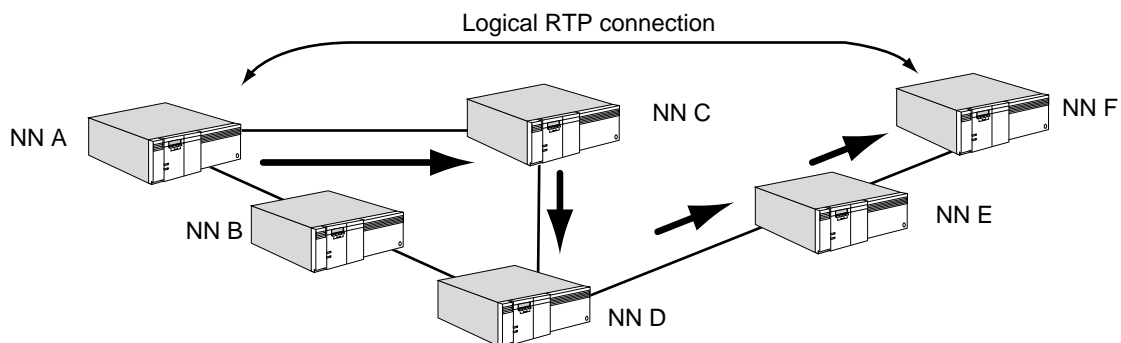
Figure 188 is an example where nondisruptive path switching takes place. In the figure, all network nodes shown are in an HPR network. There is an RTP connection between network node A and network node F, and network node C serves as an intermediate node on the path. When the link between node C and node F goes down and the connection times out, a nondisruptive path switch is triggered from network node A. Network node A uses Topology Routing Services (TRS) to determine the best alternate path. The least cost alternate path is the one that goes to network node C, then through network nodes D and E, and then to node F. The logical RTP connection remains up, even though one of the original links failed.

Figure 188 Nondisruptive Path Switching (Example)

1. RTP connection between NN A and NN F is up.



2. Link between NN C and NN F goes down. NN A tries a path switch, trying to find the best alternate path.



3. NN A determines best alternate path using TRS and conducts a path switch to the new path. RTP connection stays up.

Adaptive Rate Pacing

Adaptive-rate-based congestion and flow control is a mechanism for RTP endpoints to regulate the amount of traffic entering the HPR network. This method determines if the network is congested based on the rate of traffic entering the network, the rate of traffic leaving the network, and the buffer situation of the receiving RTP endpoint. Attempts are made to allocate equal bandwidth to all RTP connections over a link that is shared by the RTP connections. However, sessions sharing one RTP connection require individual session pacing to ensure that any one session does not occupy the whole RTP connection.

Comparison of ISR and HPR Functions

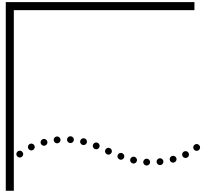
Table 31 lists APPN features supported on ISR nodes (base APPN) and features supported on HPR-capable nodes. This information applies to the 3Com implementation of both HPR and ISR.

Table 31 Comparison of ISR and HPR Function Support

| APPN Feature | Support on ISR Network Node (Base APPN) | Support on HPR Network Node |
|---|---|-----------------------------|
| DLUr support | Yes | Yes |
| APPN over SDLC links | Yes | Yes |
| Routing over WANs (not supported on ATM) | Yes | Yes |
| Parallel TGs | Yes | Yes |
| DLSw links between network nodes (for HPR, both network nodes must support HPR) | Yes | Yes |
| APPN in Boundary Routing topologies | Yes | Yes |
| APPN connection networks | Yes | Yes |
| Route calculation using class of service* | Yes | Yes |
| Rapid Transport Protocol support | No | Yes |
| Automatic Network Routing | No | Yes |
| Nondisruptive path switching | No | Yes |
| Adaptive-rate-based congestion control | No | Yes |
| Link level error recovery | Required [†] | Not required [†] |

* Route calculation operates the same for both ISR and HPR nodes. By default, no priority is given to paths between HPR nodes vs. paths between ISR nodes.

† APPN ISR uses LLC2 to provide link level error recovery. HPR provides the option of not using link level error recovery, which reduces CPU processing overhead on intermediate nodes.



CONFIGURING APPN CLASS OF SERVICE

This chapter describes the Advanced Peer-to-Peer Networking (APPN) class of service (COS) and how it is used to calculate routes in an APPN network.

The class of service database exists in all APPN network nodes and helps determine how traffic is routed within the APPN network. This database determines how sessions are routed based on such characteristics as transmission priority, security levels, line speed, propagation delay, and resistance (the desirability of routing on the node).

Levels of class of service are used because different applications have different response time and throughput requirements. For example, interactive applications (such as a session between a terminal user and a host) normally require faster data transmission and consistent response times, while batch file transfers require high throughput and are not response-time oriented.

The same method of calculating routes based on class of service is used for both Intermediate Session Routing (ISR) and High Performance Routing (HPR) traffic. However, using the default class of service tables, no special priority is given for HPR links over ISR links.

Default SNA Class of Service Modes

IBM has created a set of Systems Network Architecture (SNA)-defined mode names and corresponding class of service names that are applicable to the vast majority of user environments. When end stations issue a session request to the network node using the IBM defaults, the network node maps the mode name to one of the default classes of service. The default COS definitions are preconfigured in your bridge/router so you do not need to perform any configuration to use them.

Table 32 lists the IBM default mode names and corresponding class of service names. In the table, the pound character (#) is equivalent to the hex value X'7B' as defined in the IBM architecture documents. For more information on this value, see the IBM documents, *Systems Network Architecture Type 2.1 Node Reference* and *Systems Network Architecture LU 6.2 Reference: Peer Protocols*. For the contents of the default SNA class of service tables, see "Default Class of Service Tables" later in this chapter.

Table 32 SNA Default Mode Names and Corresponding Class of Service Names

| Mode Name | Class of Service Name | Transmission Priority |
|-------------------------------|-----------------------|-----------------------|
| blank (no characters entered) | #CONNECT | Medium |
| #BATCH | #BATCH | Low |
| #BATCHSC | #BATCHSC | Low |
| #INTER | #INTER | High |
| #INTERSC | #INTERSC | High |
| CPSVCMG | CPSVCMG | Network |
| SNASVCMG | SNASVCMG | Network |

Table 32 SNA Default Mode Names and Corresponding Class of Service Names

| Mode Name | Class of Service Name | Transmission Priority |
|-----------|-----------------------|-----------------------|
| CPSVRMGR* | SNASVCMG | Network |

* This mode is used only for the CP-SVR pipe for sessions between a DLUR and DLUs.

A session request may include a Class of Service/Transmission Priority Field (COS/TPF). If a session request includes a COS/TPF and the network node knows about it (if it is one of the IBM defaults or has been defined on the network node), the network node processes the request with the COS specified in the COS/TPF. If the network node does not know the COS, then it uses the mode name to map one. If the network node does not know about the mode name, the session request will be rejected. Some implementations default to #CONNECT if the network node does not know the mode name. If the session request does not have the COS/TPF, then the network node tries to map it; if the network node cannot map it, the session request will be rejected.

To accept nonstandard modes from the end node, a class of service name must be mapped to the mode name. For information on mapping mode names to customized class of service names and creating customized class of service tables to meet specialized needs, see “Creating Customized Class of Service Tables” next.

Creating Customized Class of Service Tables

When you use customized class of service tables, you have more flexibility in determining how your network handles load balancing among different paths. You can also set prioritization of sessions and control response time. Customized COS definitions can also be useful for larger networks handling greater numbers of sessions. If a customized class of service mode name has been created on one of your end stations, then you also want to define the class of service on the network node.

If an end station issues a session request with a nonstandard mode, then a customized class of service must be created to handle that mode. If the nonstandard mode is not defined on the network node, the network node will use the default class of service for unknown mode names, which is #CONNECT.

To add a customized class of service to the bridge/router, follow these steps:

- 1 Determine the transmission needs of the class of service, and the specific sessions the class of service will be used for.
- 2 Create the customized class of service, specifying in order the class of service name, mode name, and transmission priority using:

```
ADD -APPN ConfigCOS <cos name> <transmit priority> [SNA defined COS name]
```

If you specify an IBM-defined COS name in the command, you can automatically copy the node row and transmission group row characteristics from the IBM-defined class of service to the class of service you create. For more information on the ConfigCOS parameter syntax, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

- 3 Configure class of service node rows, specifying the class of service name and the other attributes of the node row, using:

```
ADD -APPN COSNodeRow <cos name> <weight>(0-255) [Congestion=min (Yes|No),max (Yes|No)] [Resistance=min,max]
```

For more information on COSNodeRow parameter values, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

- 4 Configure transmission group rows, specifying the class of service name and other attributes of the transmission group using:

```
ADD -APPN COSTgRow <cos name> <weight>(0-255) [ConnectCost=min,max]
    [ByteCost=min,max] [Security=min,max] [PropDelay=min,max]
    [EffectCap=min,max] [Usd1=min,max] [Us2=min,max] [Usd3=min,max]
```

For more information on COSTgRow parameter values, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

- 5 To define the newly created class of service to the system, use:

```
SET -APPN COSDef = <cos name>
```

After this command is entered, the new class of service will be used by the system.

Mapping Class of Service Names to Mode Names

To map a class of service name to one or more mode names, use:

```
ADD -APPN ModetoCosMap <cos_name> <mode_name> [mode_name ...]
```

Use this command to map any mode names to a customized COS name you have created. An incoming session with the specific mode name will be able to map the mode name to the customized COS. You can also map mode names to default SNA classes of service.

Displaying Class of Service Information

To display a list of available classes of service, enter:

```
SHow -APPN COS
```

To display a list of all class of service node rows, including IBM default node tables, enter:

```
SHow -APPN COSNodeChar
```

To display a list of all class of service transmission group rows, including IBM default transmission group (TG) tables, enter:

```
SHow -APPN COSTgChar
```

To display a list of available modes, enter:

```
SHow -APPN Mode
```

To display a list of mode names that are mapped to class of service names, enter:

```
SHow -APPN ModetoCosMap
```

For more information about these parameters, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

To display a cached tree for a class of service showing the route to a destination node, including the weight of intermediate nodes, enter:

```
SHow -APPN TreeCache
```

The display shows all classes of service in the cache. Optionally, you can specify a class of service with this command.

Deleting Class of Service Information

To delete a customized class of service table, use

```
DELeTe -APPN ConfigCOS <cos name>
```

To delete a class of service node row, enter the DELeTe -APPN COSNodeRow command and specify the class of service name and row number to be deleted.

For example, to delete node row 8 in the customized class of service "SanJose," enter:

```
DELEte -APPN COSNodeRow SanJose 8
```

To delete a class of service transmission group row, enter the DELEte -APPN COSTgRow command, and specify the class of service name and row number to be deleted. For example, to delete transmission group row 8 in the customized class of service "SanJose," enter:

```
DELEte -APPN COSTgRow SanJose 8
```

How Class of Service Calculates Routes

The APPN class of service database determines the best routing path by comparing the various factors that make some paths more desirable than others. Among the factors considered are the congestion of nodes along the path, the resistance (desirability of routing) for the nodes along the path, and the characteristics of the transmission groups (such as byte cost and connection cost) *between* the nodes along the path.

Figure 189 is an example of an APPN network. In this example, the class of service tables are used to calculate the best path between the network node in San Jose and the network node in New York. This example shows a simple scenario with a single transmission group between each node. However, two TGs (also known as parallel TGs) are supported between network nodes.

There are several possible paths. For this example, the paths are designated as follows:

Path A: San Jose→Seattle→Chicago→Philadelphia→New York

Path B: San Jose→Seattle→Chicago→Washington D.C.→New York

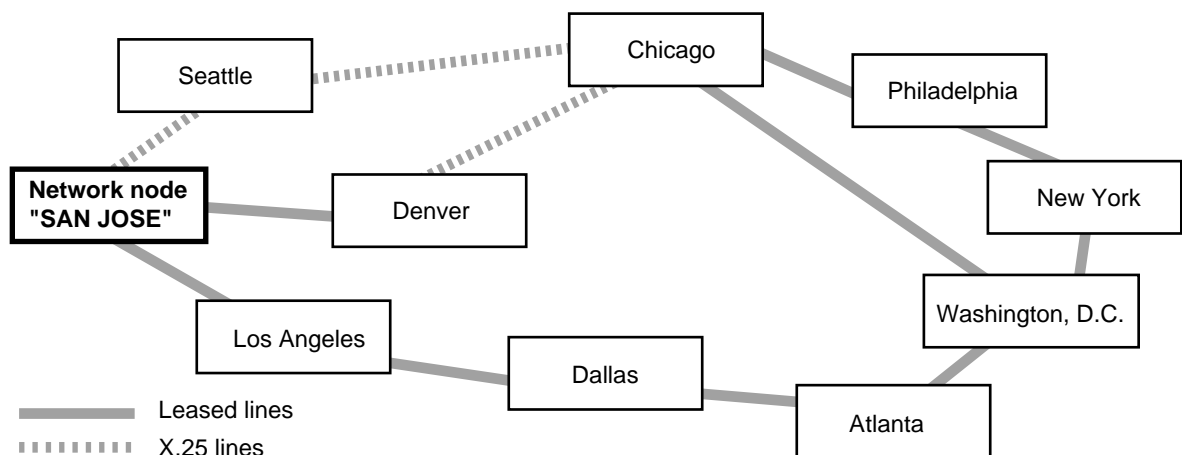
Path C: San Jose→Denver→Chicago→Philadelphia→New York

Path D: San Jose→Denver→Chicago→Washington D.C.→New York

Path E: San Jose→Los Angeles→Dallas→Atlanta→Washington D.C.→New York

All nodes in the example are APPN network nodes.

Figure 189 COS Example (Network Topology)



Step 1: Determining Node Weights Along a Path

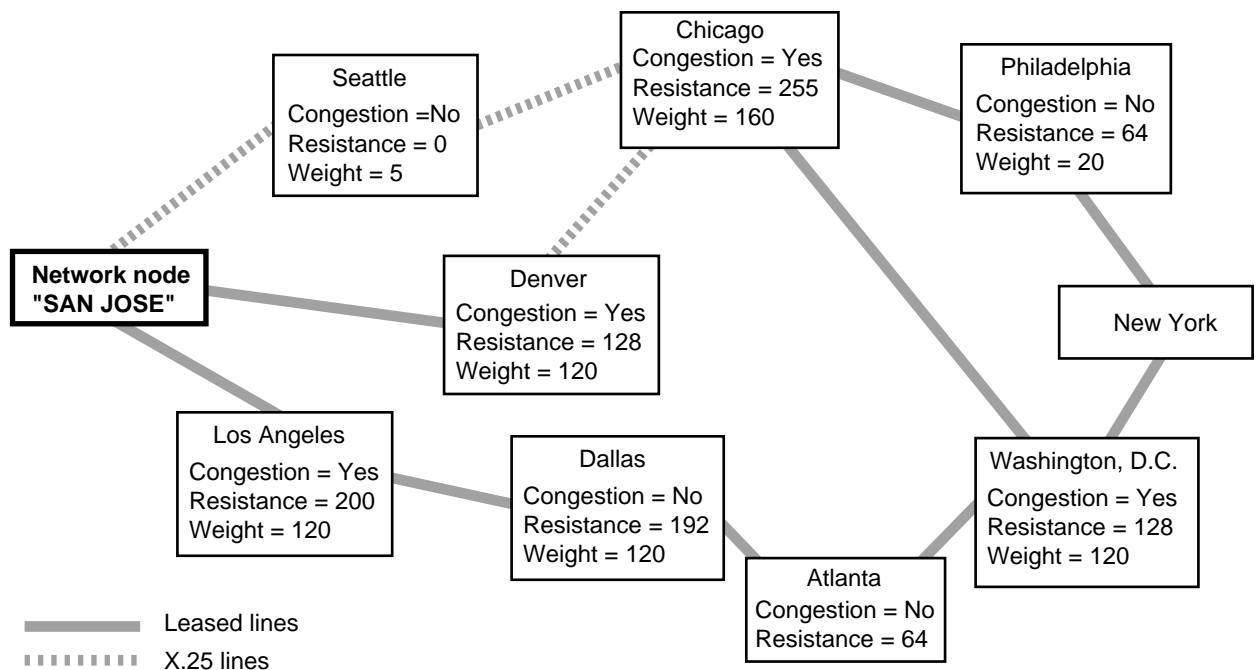
The first step in how routing and topology services determine the best path is by adding the weight of the nodes along the path. The weight of the individual nodes is determined by calculating such factors as node congestion and the resistance (desirability of routing) for each node.

Figure 190 shows the node characteristics of the nodes in the network. The figure shows the congestion and resistance values set for each node. The weight shown for each node is calculated by adding the relative factors of congestion and resistance. The lower the resistance, the more desirable the node is to route traffic through. For example, a resistance value of 0 indicates the node is highly desirable to route traffic through, a value of 128 indicates the median, meaning the node is neither highly desirable nor highly undesirable. A resistance value of 255 indicates the node is not desirable to route through.

The resistance plus the congestion value indicates the relative weight of the node. The lower the node weight, the more desirable the node is to route traffic through.

For example, the node in Seattle is uncongested while it has a resistance of 0, indicating it is a desirable node to route traffic through. In contrast, the node in Los Angeles is congested and has a rate of 200, indicating the node is less desirable for routing traffic through.

Figure 190 COS Example (Calculating Node Weights)



The weight for a given path is calculated by determining the requirements of each path. The requirements are then measured against the class of service node table. The weight of the first node row that meets the requirement of the node is

assigned to that node. To check the default node table for the IBM-defined class of service named #CONNECT, see Table 39.

To calculate the weight of a node, the resistance and congestion levels of that node are checked. The node table is then checked to determine the first node row in the table that would accept the requirements of that node; the weight assigned to the node is the weight of that node row. The lower the node row, the lower the weight assigned to the row; the lower the weight, the greater precedence that row has.

For example, the node in Denver has a resistance of 128 and is congested. A network node is congested if it has reached 90 percent of the maximum number of ISR sessions configured for that node. In the node row table, a node is considered either congested ("yes") or uncongested ("no").

When the node row table is used, each row is checked to find the first row that will accept the conditions. The process is as follows:

- 1 Node row 1 is checked. The conditions are not satisfied because the maximum resistance allowed is 31.
- 2 Node row 2 is checked. The conditions are not satisfied because the maximum resistance allowed is 63.
- 3 Node rows 3 and 4 are checked and are also rejected because the maximum resistance values allowed are still lower than Denver's resistance value of 128.
- 4 Node row 5 is checked, and because the maximum resistance allowed is 159, this is the first row that will accept all the conditions. Because the weight of row 5 is 60, that is the weight assigned to the Denver node.



In this example, if the Denver node were congested, the first node row that would satisfy all conditions would be row 7, which would then assign a weight of 120, changing the total weight of the path.

Using this formula, the appropriate weights of each node are calculated. Table 33 lists the correct weights for each node in the figure based on this class of service mode table. (If a different class of service mode is used, a different node table is used, which changes the various calculations.)

Table 33 Node Weights Based on Node Row Formula (Example)

| Node | Weight Based on COS Node Row (for IBM-default COS #CONNECT) |
|-----------------|---|
| Seattle | 5 |
| Denver | 60 |
| Los Angeles | 120 |
| Chicago | 160 |
| Dallas | 120 |
| Atlanta | 20 |
| Philadelphia | 20 |
| Washington D.C. | 120 |

After the weight of each node is determined, then the weights of all nodes on a path are added together; this determines the total node weight of a given path.

Based on the weight calculations in Table 33, the total node weight of each path is shown in Table 34.

Table 34 Total Node Weight for Each Path (Example)

| Path | Total Node Weight |
|--------|-------------------|
| PATH A | 185 |
| PATH B | 285 |
| PATH C | 240 |
| PATH D | 340 |
| PATH E | 380 |

The table indicates that of the four paths, path A has the lowest weight, which does not mean that path A is the best path. Calculating the weight of the nodes along a path is only the first step. The weights of the transmission groups for each path are then calculated. Proceed to the next section.

Step 2: Determining TG Weights Along a Path

The second factor determining the weight of a path is the weight of all the TGs along the path. The TG consists of the path between two adjacent network nodes. The number of TGs on a path is determined by the number of network nodes on the path; the more nodes on the path, the more TGs there are on the path. For example, on Path A, there are four TGs from the San Jose node to the New York node. On Path E, there are five TGs because Path E includes an additional node. Figure 191 shows the different transmission groups.

Figure 191 COS Example (Configuring TG Weights)

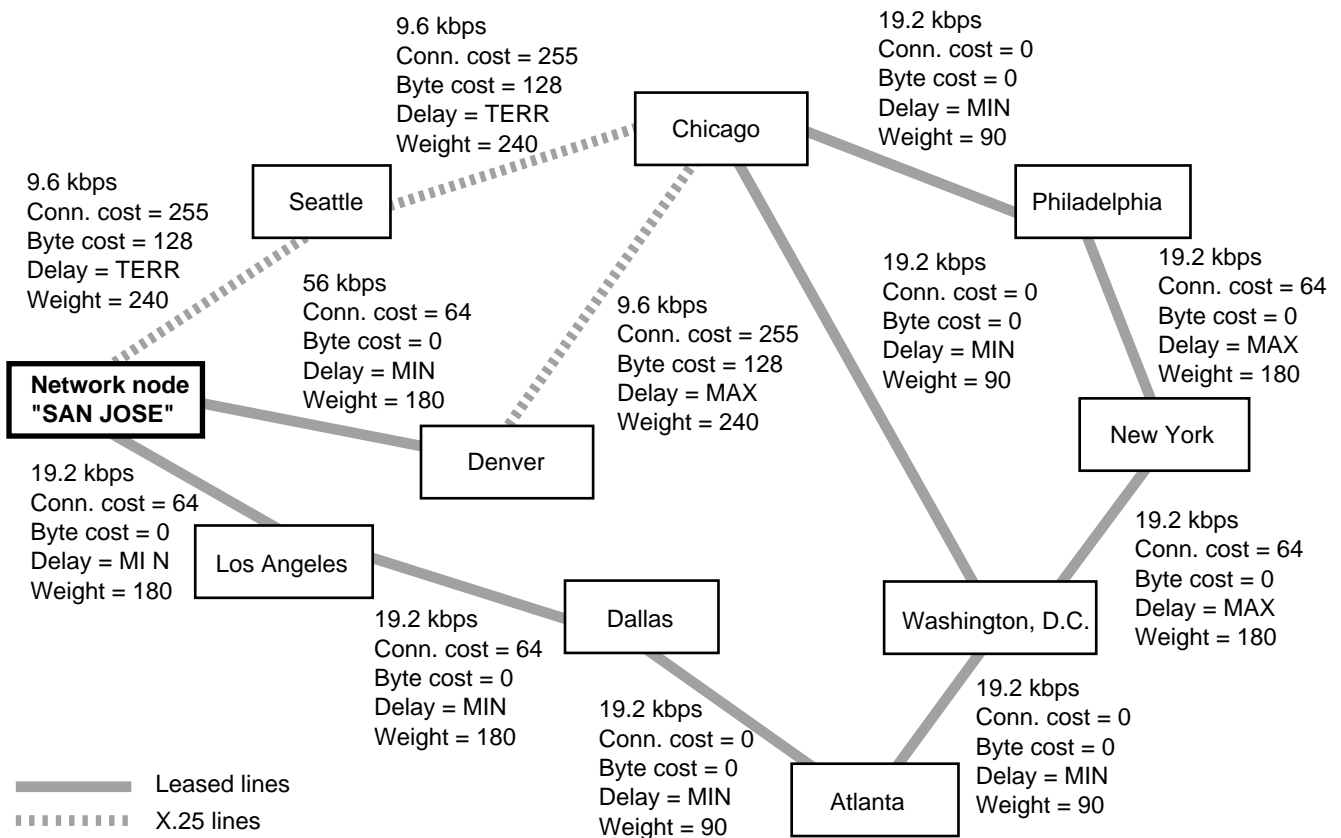


Table 35 lists the same information contained in the figure. It lists the attributes of each TG used in calculating the weight of each TG. These attributes are for the example only. No user-defined parameters are used in the example.

Table 35 TG Attributes Example

| Transmission Group (Link Between Two Network Nodes) | Conn. Cost | Byte Cost | Prop. Delay | Encode Capacity | Security |
|--|-------------------|------------------|--------------------|------------------------|-----------------|
| San Jose→Seattle | 255 | 128 | TERR | 9,600 | MINIMAL |
| San Jose→Denver | 64 | 0 | MIN | 56,000 | MINIMAL |
| San Jose→Los Angeles | 64 | 0 | MIN | 19,200 | MINIMAL |
| Seattle→Chicago | 255 | 128 | TERR | 9,600 | MINIMAL |
| Denver→Chicago | 255 | 128 | MAX | 9,600 | MINIMAL |
| Los Angeles→Dallas | 64 | 0 | MIN | 19,200 | MINIMAL |
| Chicago→Philadelphia | 0 | 0 | MIN | 19,200 | MINIMAL |
| Chicago→Washington D.C. | 0 | 0 | MIN | 19,200 | MINIMAL |
| Dallas→Atlanta | 0 | 0 | MIN | 19,200 | MINIMAL |
| Atlanta→Washington D.C. | 0 | 0 | MIN | 19,200 | MINIMAL |
| Philadelphia→New York | 64 | 0 | MAX | 19,200 | MINIMAL |
| Washington D.C.→New York | 64 | 0 | MAX | 19,200 | MINIMAL |

To determine the weight of each TG, the class of service TG table is checked. The first row in the TG table that meets the requirements of that TG is used to calculate the weight of the TG.

For example, the TG between San Jose and Seattle has a connection cost of 255 and a byte cost of 128. It has an encoding capacity of 9,600. Table 39 shows the default TG values for the default class of service "#CONNECT."

When the TG row table is used, each row is checked to find the first row that will accept the conditions. The process is as follows:

- 1 TG row 1 is checked. The conditions are not satisfied because both the connection cost and byte cost exceed the maximum in TG row 1. (If only one of the attributes exceeded the maximum, that would have been enough to reject TG row 1.)
- 2 TG rows 2 through 5 are checked and are rejected because the TG's connection cost and byte cost exceed the maximums in those rows.
- 3 TG row 6 is checked, and the TG's byte cost of 128 matches the maximum allowed in the TG row. The row does not satisfy all the conditions because the TG's connection cost is 255, and the maximum connection cost allowed in row 6 is 128.
- 4 TG row 7 is checked and is again rejected because the maximum connection cost allowed is not high enough.
- 5 TG row 8 is checked, and because it allows a maximum connection cost of 255, TG row 8 is the row assigned to the TG. Because the weight for TG row 8 is 240, this is the weight assigned for the TG between San Jose and Seattle.

Using Table 40 and the checking process, the weight for each TG is calculated. Table 36 lists the weights calculated based on this class of service mode. (If a

different class of service mode is used, a different TG table is used, which changes the various calculations.)

Table 36 TG Weights Based on Default Class of Service TG Table

| Transmission Group | Weight Based on COS TG Row (for IBM-default COS #CONNECT) |
|--------------------------|---|
| San Jose→Seattle | 240 |
| San Jose→Denver | 180 |
| San Jose→Los Angeles | 180 |
| Seattle→Chicago | 240 |
| Denver→Chicago | 240 |
| Los Angeles→Dallas | 180 |
| Chicago→Philadelphia | 90 |
| Chicago→Washington D.C. | 90 |
| Dallas→Atlanta | 90 |
| Atlanta→Washington D.C. | 90 |
| Philadelphia→New York | 180 |
| Washington D.C.→New York | 180 |

After the weights of all the TGs are calculated, the weights of the four paths are calculated by adding the weights of each TG on the path. Table 37 lists the total TG weights for the four paths in the example.

Table 37 Total TG Weight for Each Path (Example)

| Path | Total TG Weight |
|--------|-----------------|
| PATH A | 750 |
| PATH B | 750 |
| PATH C | 690 |
| PATH D | 690 |
| PATH E | 720 |

After the total TG weight for each path is calculated, this total TG weight is added to the total node weight to calculate the total weight for each path. Proceed to the next section.

Step 3: Calculating the Total Weight for Each Path

To calculate the total weight of each path, the total node weight is added to the total TG weight. Table 38 lists the weight values for the four paths.

Table 38 Total Calculated Weight for Each Path (Example)

| Path | Total Node Weight + | Total TG Weight = | Total Weight for Each Path |
|--------|---------------------|-------------------|----------------------------|
| PATH A | 185 | 750 | 935 |
| PATH B | 285 | 750 | 1035 |
| PATH C | 240 | 690 | 930 |
| PATH D | 340 | 690 | 1030 |
| PATH E | 380 | 720 | 1100 |

After calculating the total path weight, the class of service determines that the best route from the network node in San Jose to the network node in New York is

Path C (San Jose→Denver→Chicago→Philadelphia→New York) because Path C has the lowest weight.



Dynamic network conditions can affect the weight of a node. For example, if a node that is congested becomes uncongested, then the weight of the node will be lower. The changed node weight will affect the calculation of total path weights and change which is the best route. The APPN class of service calculates the best route at the time of the session request.

Default Class of Service Tables

This section lists the default SNA class of service tables that are used for calculating routes. In all tables, the user-defined values are not shown; the minimum user-defined value is 0 and the maximum is 255.

Default Node Table

Table 39 lists the default node table that applies to the different modes. The same node table is used regardless of the mode; it is the mode that determines the transmission priority that differentiates the calculation for node tables. See Table 32 for a list of the default modes and corresponding class of service names.

Table 39 Node Table for Default Classes of Service

| Row Number | Weight | Congestion | | Node Resistance |
|------------|--------|------------|-----|-----------------|
| 1 | 5 | Min. | No | 0 |
| | | Max. | No | 31 |
| 2 | 10 | Min. | No | 0 |
| | | Max. | No | 63 |
| 3 | 20 | Min. | No | 0 |
| | | Max. | No | 95 |
| 4 | 40 | Min. | No | 0 |
| | | Max. | No | 127 |
| 5 | 60 | Min. | No | 0 |
| | | Max. | No | 159 |
| 6 | 80 | Min. | No | 0 |
| | | Max. | No | 191 |
| 7 | 120 | Min. | No | 0 |
| | | Max. | Yes | 223 |
| 8 | 160 | Min. | No | 0 |
| | | Max. | Yes | 255 |

Default TG Tables

This section lists the default TG tables for each class of service.

Table 40 lists the default TG table for the default class of service #CONNECT. The corresponding mode name is blank (that is, no characters are entered), and the transmission priority is medium.

Table 40 #CONNECT Default Class of Service TG Table

| Row Number | Weight | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity | |
|------------|--------|------------|-----------|----------|-------------|-----------------|---------|
| 1 | 30 | Min. | 0 | 0 | MINIMAL | MIN | 0x76 |
| | | Max. | 0 | 0 | RAD_GUARD | NEGL | MAXIMUM |
| 2 | 60 | Min. | 0 | 0 | MINIMAL | MIN | 56000 |
| | | Max. | 0 | 0 | RAD_GUARD | TERR | MAXIMUM |

Table 40 #CONNECT Default Class of Service TG Table (continued)

| Row Number | Weight | | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|------|------------|-----------|-----------|-------------|-----------------|
| 3 | 90 | Min. | 0 | 0 | MINIMAL | MIN | 19200 |
| | | Max. | 0 | 0 | RAD_GUARD | TERR | MAXIMUM |
| 4 | 120 | Min. | 0 | 0 | MINIMAL | MIN | 9600 |
| | | Max. | 0 | 0 | RAD_GUARD | TERR | MAXIMUM |
| 5 | 150 | Min. | 0 | 0 | MINIMAL | MIN | 19200 |
| | | Max. | 0 | 0 | RAD_GUARD | PKT | MAXIMUM |
| 6 | 180 | Min. | 0 | 0 | MINIMAL | MIN. | 9600 |
| | | Max. | 128 | 128 | RAD_GUARD | PKT | MAXIMUM |
| 7 | 210 | Min. | 0 | 0 | MINIMAL | MIN | 4800 |
| | | Max. | 196 | 196 | RAD_GUARD | MAX | MAXIMUM |
| 8 | 240 | Min. | 0 | 0 | MINIMAL | MIN | 0x00 |
| | | Max. | 255 | 255 | RAD_GUARD | MAX | MAXIMUM |

Table 41 lists the default TG table for the default class of service #BATCH. The corresponding mode name is #BATCH, and the transmission priority is low.

Table 41 #BATCH Default Class of Service TG Table

| Row Number | Weight | | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|------|------------|-----------|-----------|-------------|-----------------|
| 1 | 30 | Min. | 0 | 0 | MINIMAL | MIN | 57 |
| | | Max. | 0 | 0 | RAD_GUARD | NEGL | 603979776 |
| 2 | 60 | Min. | 0 | 0 | MINIMAL | MIN | 19 |
| | | Max. | 0 | 0 | RAD_GUARD | TERR | 603979776 |
| 3 | 90 | Min. | 0 | 0 | MINIMAL | MIN | 19 |
| | | Max. | 128 | 128 | RAD_GUARD | TERR | 603979776 |
| 4 | 120 | Min. | 0 | 0 | MINIMAL | MIN | 9 |
| | | Max. | 0 | 0 | RAD_GUARD | TERR | 603979776 |
| 5 | 150 | Min. | 0 | 0 | MINIMAL | MIN | 9 |
| | | Max. | 128 | 128 | RAD_GUARD | PKT | 603979776 |
| 6 | 180 | Min. | 0 | 0 | MINIMAL | MIN. | 9 |
| | | Max. | 0 | 0 | RAD_GUARD | PKT | 603979776 |
| 7 | 210 | Min. | 0 | 0 | MINIMAL | MIN | 4 |
| | | Max. | 196 | 196 | RAD_GUARD | MAX | 603979776 |
| 8 | 240 | Min. | 0 | 0 | MINIMAL | MIN | 0 |
| | | Max. | 255 | 255 | RAD_GUARD | MAX | 603979776 |

Table 42 lists the default TG table for the default class of service #BATCHSC. The corresponding mode name is #BATCHSC, and the transmission priority is low.

Table 42 #BATCHSC Default Class of Service TG Table

| Row Number | Weight | | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|------|------------|-----------|------------|-------------|-----------------|
| 1 | 30 | Min. | 0 | 0 | PUB_SWITCH | MIN | 57 |
| | | Max. | 0 | 0 | RAD_GUARD | NEGL | 603979776 |
| 2 | 60 | Min. | 0 | 0 | PUB_SWITCH | MIN | 19 |
| | | Max. | 0 | 0 | RAD_GUARD | TERR | 603979776 |
| 3 | 90 | Min. | 0 | 0 | PUB_SWITCH | MIN | 19 |
| | | Max. | 128 | 128 | RAD_GUARD | TERR | 603979776 |
| 4 | 120 | Min. | 0 | 0 | PUB_SWITCH | MIN | 9 |
| | | Max. | 0 | 0 | RAD_GUARD | TERR | 603979776 |

Table 42 #BATCHSC Default Class of Service TG Table

| Row Number | Weight | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|------------|-----------|------------|-------------|-----------------|
| 5 | 150 | Min. 0 | 0 | PUB_SWITCH | MIN | 9 |
| | | Max. 128 | 128 | RAD_GUARD | PKT | 603979776 |
| 6 | 180 | Min. 0 | 0 | PUB_SWITCH | MIN. | 9 |
| | | Max. 0 | 0 | RAD_GUARD | PKT | 603979776 |
| 7 | 210 | Min. 0 | 0 | PUB_SWITCH | MIN | 4 |
| | | Max. 196 | 196 | RAD_GUARD | MAX | 603979776 |
| 8 | 240 | Min. 0 | 0 | PUB_SWITCH | MIN | 0 |
| | | Max. 255 | 255 | RAD_GUARD | MAX | 603979776 |

Table 43 lists the default TG table for the default class of service #INTER. The corresponding mode name is #INTER, and the transmission priority is high.

Table 43 #INTER Default Class of Service TG Table

| Row Number | Weight | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|------------|-----------|-----------|-------------|-----------------|
| 1 | 30 | Min. 0 | 0 | MINIMAL | MIN | 4300 |
| | | Max. 0 | 0 | RAD_GUARD | NEGL | 603979776 |
| 2 | 60 | Min. 0 | 0 | MINIMAL | MIN | 57 |
| | | Max. 0 | 0 | RAD_GUARD | TERR | 603979776 |
| 3 | 90 | Min. 0 | 0 | MINIMAL | MIN | 57 |
| | | Max. 128 | 128 | RAD_GUARD | TERR | 603979776 |
| 4 | 120 | Min. 0 | 0 | MINIMAL | MIN | 19 |
| | | Max. 0 | 0 | RAD_GUARD | TERR | 603979776 |
| 5 | 150 | Min. 0 | 0 | MINIMAL | MIN | 19 |
| | | Max. 128 | 128 | RAD_GUARD | PKT | 603979776 |
| 6 | 180 | Min. 0 | 0 | MINIMAL | MIN. | 9 |
| | | Max. 0 | 0 | RAD_GUARD | PKT | 603979776 |
| 7 | 210 | Min. 0 | 0 | MINIMAL | MIN | 9 |
| | | Max. 196 | 196 | RAD_GUARD | MAX | 603979776 |
| 8 | 240 | Min. 0 | 0 | MINIMAL | MIN | 0 |
| | | Max. 255 | 255 | RAD_GUARD | MAX | 603979776 |

Table 44 lists the default TG table for the default class of service #INTERSC. The corresponding mode name is #INTERSC, and the transmission priority is high.

Table 44 #INTERSC Default Class of Service TG Table

| Row Number | Weight | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|------------|-----------|------------|-------------|-----------------|
| 1 | 30 | Min. 0 | 0 | PUB_SWITCH | MIN | 4300 |
| | | Max. 0 | 0 | RAD_GUARD | NEGL | 603979776 |
| 2 | 60 | Min. 0 | 0 | PUB_SWITCH | MIN | 57 |
| | | Max. 0 | 0 | RAD_GUARD | TERR | 603979776 |
| 3 | 90 | Min. 0 | 0 | PUB_SWITCH | MIN | 57 |
| | | Max. 128 | 128 | RAD_GUARD | TERR | 603979776 |
| 4 | 120 | Min. 0 | 0 | PUB_SWITCH | MIN | 19 |
| | | Max. 0 | 0 | RAD_GUARD | TERR | 603979776 |

Table 44 #INTERSC Default Class of Service TG Table

| Row Number | Weight | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|--------------------|-----------|-------------------------|-------------|-----------------|
| 5 | 150 | Min. 0
Max. 128 | 0
128 | PUB_SWITCH
RAD_GUARD | MIN
PKT | 19
603979776 |
| 6 | 180 | Min. 0
Max. 0 | 0
0 | PUB_SWITCH
RAD_GUARD | MIN.
PKT | 9
603979776 |
| 7 | 210 | Min. 0
Max. 196 | 0
196 | PUB_SWITCH
RAD_GUARD | MIN
MAX | 9
603979776 |
| 8 | 240 | Min. 0
Max. 255 | 0
255 | PUB_SWITCH
RAD_GUARD | MIN
MAX | 0
603979776 |

Table 45 lists the default TG table for the default class of service CPSVCMG. The corresponding mode name is CPSVCMG, and the transmission priority is network.

Table 45 CPSVCMG Default Class of Service TG Table

| Row Number | Weight | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|--------------------|-----------|----------------------|-------------|-------------------|
| 1 | 30 | Min. 0
Max. 0 | 0
0 | MINIMAL
RAD_GUARD | MIN
NEGL | 4300
603979776 |
| 2 | 60 | Min. 0
Max. 0 | 0
0 | MINIMAL
RAD_GUARD | MIN
TERR | 57
603979776 |
| 3 | 90 | Min. 0
Max. 0 | 0
0 | MINIMAL
RAD_GUARD | MIN
TERR | 9
603979776 |
| 4 | 120 | Min. 0
Max. 0 | 0
0 | MINIMAL
RAD_GUARD | MIN
TERR | 9
603979776 |
| 5 | 150 | Min. 0
Max. 0 | 0
0 | MINIMAL
RAD_GUARD | MIN
PKT | 19
603979776 |
| 6 | 180 | Min. 0
Max. 128 | 0
128 | MINIMAL
RAD_GUARD | MIN.
MAX | 9
603979776 |
| 7 | 210 | Min. 0
Max. 196 | 0
196 | MINIMAL
RAD_GUARD | MIN
MAX | 4
603979776 |
| 8 | 240 | Min. 0
Max. 255 | 0
255 | MINIMAL
RAD_GUARD | MIN
MAX | 0
603979776 |

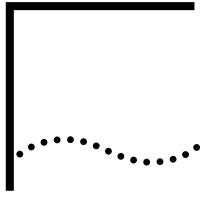
Table 46 lists the default TG table for the default class of service SNASVCMG. The corresponding mode name is either SNASVCMG or CPSVRMG, and the transmission priority in both cases is network.

Table 46 SNASVCMG Default Class of Service TG Table

| Row Number | Weight | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|------------------|-----------|----------------------|-------------|-------------------|
| 1 | 30 | Min. 0
Max. 0 | 0
0 | MINIMAL
RAD_GUARD | MIN
NEGL | 4300
603979776 |
| 2 | 60 | Min. 0
Max. 0 | 0
0 | MINIMAL
RAD_GUARD | MIN
TERR | 57
603979776 |
| 3 | 90 | Min. 0
Max. 0 | 0
0 | MINIMAL
RAD_GUARD | MIN
TERR | 19
603979776 |

Table 46 SNASVCMG Default Class of Service TG Table

| Row Number | Weight | | Conn. Cost | Byte Cost | Security | Prop. Delay | Encode Capacity |
|------------|--------|------|------------|-----------|-----------|-------------|-----------------|
| 4 | 120 | Min. | 0 | 0 | MINIMAL | MIN | 9 |
| | | Max. | 0 | 0 | RAD_GUARD | TERR | 603979776 |
| 5 | 150 | Min. | 0 | 0 | MINIMAL | MIN | 19 |
| | | Max. | 0 | 0 | RAD_GUARD | PKT | 603979776 |
| 6 | 180 | Min. | 0 | 0 | MINIMAL | MIN. | 9 |
| | | Max. | 128 | 128 | RAD_GUARD | PKT | 603979776 |
| 7 | 210 | Min. | 0 | 0 | MINIMAL | MIN | 4 |
| | | Max. | 196 | 196 | RAD_GUARD | MAX | 603979776 |
| 8 | 240 | Min. | 0 | 0 | MINIMAL | MIN | 0 |
| | | Max. | 255 | 255 | RAD_GUARD | MAX | 603979776 |



CONFIGURING IPX ROUTING

This chapter describes the procedures for configuring your system to perform Internetwork Packet Exchange (IPX) routing. It also describes how the router works and gives guidelines for operating, managing, and troubleshooting it.



For conceptual information, see "How the IPX Router Works" later in this chapter.

Setting Up a Basic IPX Router

Use the following procedures to set up your system to route IPX packets. After you complete the procedures in this section, verify that the system is routing packets properly using the procedures in "Verifying the Configuration" later in this chapter.

Configuring for Local Area Networks and Point-to-Point Links

Use this procedure to configure basic IPX routing over LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router using the procedure in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Procedure

To configure the basic IPX router for LANs and PPP links, follow these steps:

- 1 Configure the network number connected through each router interface using:

```
SETDefault !<port> -IPX NETnumber = &<number>(0-FFFFFFFD) [Ethernet | Ieee  
| Llc | Snap | PPP]
```

Valid network numbers consist of up to eight hexadecimal digits in the range &0 to &FFFFFFFD. The network numbers &FFFFFFE and &FFFFFFF are reserved. You do not have to specify leading zeros in this network number.

You may also configure the port as unnumbered PPP. See the Configuring IP Routing chapter for more information.

- 2 Verify dynamic route learning is enabled using:

```
SHow !<port> -NRIP CONTrol
```

By default, NetWare Routing Information Protocol (NRIP) is set to Auto. If you do not enable NetWare Link Services Protocol (NLSP) as the routing protocol, Auto means both Talk and Listen. If NLSP is enabled, Auto means Talk if there are non-NLSP routers detected. The router is constantly listening.

When NRIP is listening, the router receives Routing Information Protocol (RIP) broadcasts and can maintain the routing table. When NRIP is talking, the router can send RIP broadcasts.

- 3 Enable IPX routing for each port using:

```
SETDefault !<port> -IPX CONTrol = ROute
```

- 4 If there are more users to serve than a primary server is licensed to handle and there is a backup server available, specify a preferred backup server using:

```
ADD !<port> -SAP PreferredServer "<server name>", ["<server name>"...]
```

After a list of preferred servers is configured, the IPX router responds to "get nearest server" requests with one of the reachable preferred servers regardless of the server location or number of hops. If no preferred server is available, the normal selection process of the nearest server takes place. In this way, the primary server and backup server can alternately serve all the users and lessen the burden on the primary server.

NetWare 4.0 clients and pre-4.0 clients specify different service types in their "get nearest server" requests. Pre-4.0 clients use File Server type (0x0004) while 4.0 clients are looking for Directory Name Server type (0x026B); appropriate preferred servers must be added.

- 5 Verify the IPX configuration by entering:

```
SHow -IPX CONFIguration
```

The router displays the IPX configuration information. If the -IPX CONTrol parameter is not set to ROute, if the network numbers are incorrect, or if the -NRIP and -SAP CONTrol parameters are not set to Talk and Listen for each port you are configuring, repeat steps 2, 3, and 4. Additional verification steps are provided in "Verifying the Configuration" later in this chapter.

To complete the configuration for PPP links, see the Configuring Wide Area Networking Using PPP chapter.

Configuring Secondary Networks with Different Header Formats

For LAN interfaces, IPX allows one physical network to be segmented into different logical networks, or secondary networks, and configured with different header formats. The header formats correspond to different encapsulation methods that allow the IPX protocol to deliver IPX packets. Table 47 lists the header formats supported by IPX encapsulation and the values associated with these formats.

Table 47 IPX Packet Header Formats

| Values | Header Formats Supported under IPX Encapsulation |
|----------|---|
| leee | IPX packets are encapsulated in IEEE 802.3 header format (Ethernet and FDDI). |
| Ethernet | IPX packets are encapsulated in Ethernet V2 header format (Ethernet only). |
| Snap | IPX packets are encapsulated in SNAP header format. |
| Llc | IPX packets are encapsulated in IEEE 802.2 header format. |



3Com recommends using Ethernet V2 for Ethernet and SNAP for FDDI and token ring.

The number of secondary networks differs between interface types:

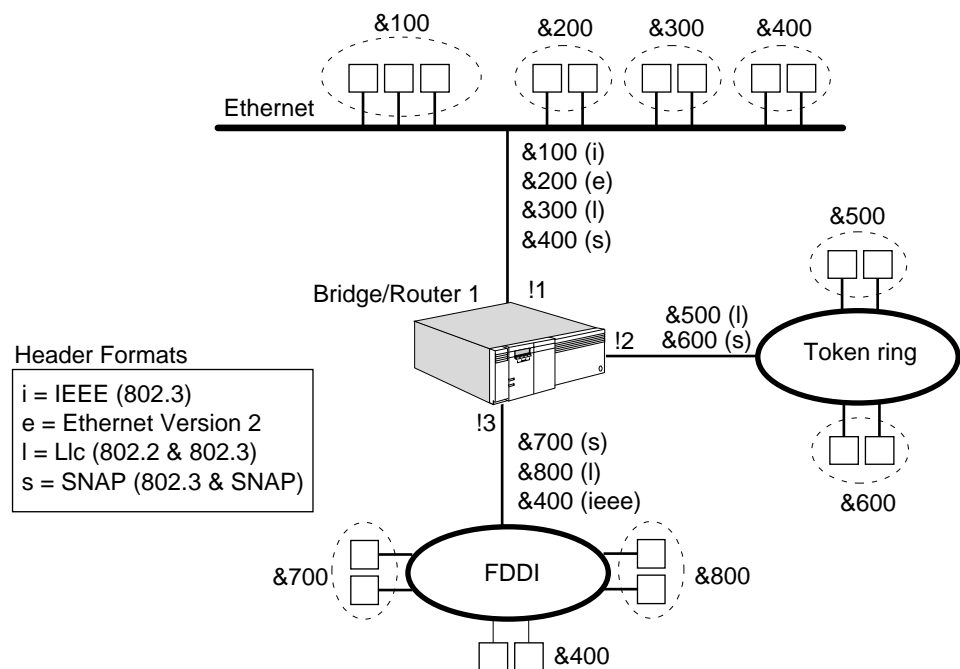
- For Ethernet interfaces, four different networks can be configured: Ethernet V2 headers (identified as Ethernet_II on Novell servers); IEEE headers (802.3 raw), identified as Ethernet_802.3 on Novell servers, and the default for Ethernet; Logical Link Control (LLC) headers (identified as Ethernet_802.2 on Novell servers); and SubNetwork Access Protocol (SNAP), identified as Ethernet_SNAP on Novell servers.
- For token ring, three different networks can be configured: LLC, SNAP, and IEEE.
- For the Fiber Distributed Data Interface (FDDI), three different networks can be configured: LLC, SNAP, and IEEE.

For each of the interface types, configure the primary network with the SETDefault command; configure the secondary networks with the ADD command.

Figure 192 shows a router with three LAN ports of different types:

- Port 1 (Ethernet) is connected to network 100 with IEEE header format, network 200 with Ethernet header format, network 300 with LLC header format, and network 400 with SNAP header format.
- Port 2 (token ring) is connected to two networks: network 500 with LLC header format and network 600 with SNAP header format.
- Port 3 (FDDI) is connected to two networks: network 700 with SNAP header format and network 800 with LLC header format.

Figure 192 Configuring Multiple Networks for Different Header Formats



To configure the primary and secondary network for port 1 shown in Figure 192, follow these steps:

- 1 Configure the primary network for port 1 by entering:

```
SETDefault !1 -IPX NETnumber = 100 Ieee
```

The primary networks for ports 2 and 3 are configured using the SETDefault command, the appropriate port number, and the appropriate header format specifier (LLC for network 500 on port 2 and SNAP for network 700 on port 3).

- 2 Configure the Ethernet secondary network for port 1 by entering:

```
ADD !1 -IPX NETnumber = 200 Ethernet
```

The remaining secondary networks for port 1 are configured using the ADD command, the port specifier !1, and the appropriate header format specifier (LLC for network 300 and SNAP for network 400).

The remaining secondary networks for ports 2 and 3 are configured using the ADD command, the appropriate port number, and the appropriate header format specifier (SNAP for network 600 on port 2 and LLC for network 800 on port 3).

Configuring for Wide Area Networks

Routing IPX over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), X.25, and ATM is supported over fully meshed, partially meshed, and nonmeshed topologies.

If you plan to route IPX over Frame Relay, ATM DXI, X.25, or ATM in a partially meshed or nonmeshed topology, you must be sure that the next-hop split horizon feature is enabled by configuring neighbors. For complete information on configuring IPX routing over Frame Relay, ATM DXI, X.25, or ATM, including a discussion on fully meshed, partially meshed, and nonmeshed topologies and next-hop split horizon, see the Configuring Wide Area Networking Using Frame Relay chapter, the Configuring Wide Area Networking Using the ATM DXI chapter, the Configuring Wide Area Networking Using X.25 chapter, or the Configuring Internetworking Using ATM chapter.

Routing IPX over SMDS is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your IPX router to perform routing over SMDS, see the Configuring Wide Area Networking Using SMDS chapter.

Nonmeshed topology may be used with virtual ports. To configure IPX routing over PPP, see the Configuring Wide Area Networking Using PPP chapter.

For WAN interfaces, you do not need to specify a header format. The formats are as follows:

- PPP uses the PPP header format.
- X.25 uses the X.25 header format.
- SMDS uses the SMDS header format.
- Frame Relay uses the Frame Relay header format.
- ATM uses the ATM header format.

You can assign secondary networks on WAN interfaces, but the status of those networks will be down.

Configuring IPXWAN over PPP

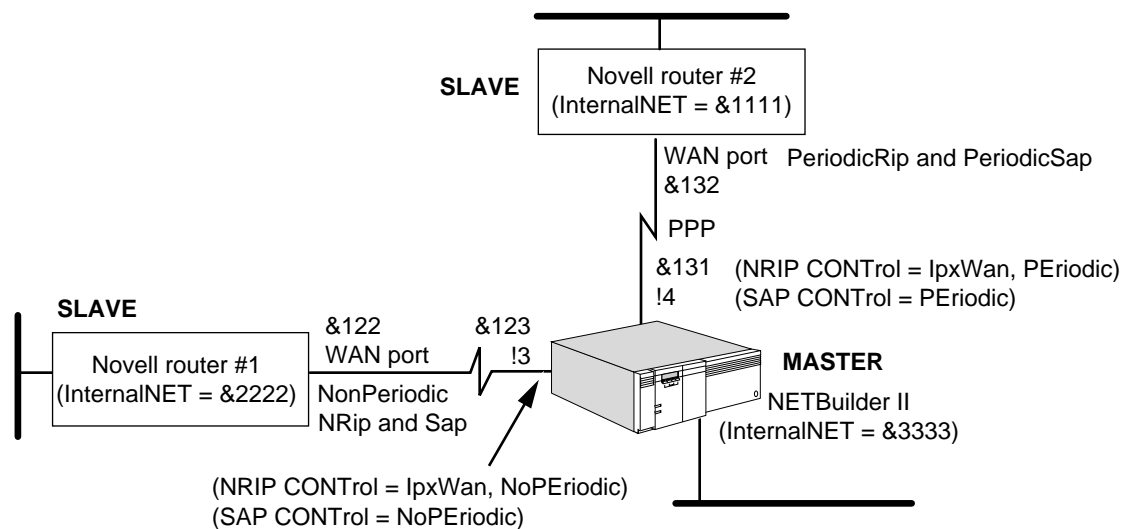
Novell has published a specification for IPX communications over wide area network services (such as PPP, X.25, Frame Relay) called IPXWAN. The specification outlines how IPX negotiations take place in these environments; for example, Novell IPX uses IPXWAN to exchange necessary router-to-router information before exchanging IPX NRIP, Service Advertising Protocol (SAP), and NLSP information over various WAN links. The 3Com implementation of the IPXWAN Protocol currently supports PPP, Frame Relay, and X.25.

To achieve interoperability between a 3Com bridge/router and a Novell Multi-Protocol Router (MPR) across a WAN link, you must configure IPXWAN over PPP on your bridge/router as shown in Figure 193.



If you are using the nonperiodic mode of NRIP and SAP, both sides of the WAN link must be configured the same way.

Figure 193 IPXWAN over PPP Using NRIP and SAP



Prerequisites

Before beginning this procedure, perform the following steps:

- Configure IPX routing on the LAN ports as described in “Configuring for Local Area Networks and Point-to-Point Links” earlier in this chapter.
- Configure PPP over the port as described in the Configuring Wide Area Networking Using PPP chapter.

Procedure

To configure IPXWAN over PPP, follow these steps:

- 1 Configure the network numbers on the wide area interfaces that will be running IPXWAN using:

```
SETDefault !<port> -IPX NETnumber = &<number>(0-FFFFFFFD)
[Ethernet | Ieee | Llc | Snap | X25 | PPP | Frame]
```

Valid network numbers consist of up to eight hexadecimal digits in the range &0 to &FFFFFFFD. The network number &FFFFFFFE and &FFFFFFF are reserved. You do not have to specify leading zeros in this network number.

- 2 Assign an internal network number to each router.

To assign the internal network number, use:

```
SETDefault -IPX InternalNET = &<number>(0-FFFFFFFD)
```

The InternalNET number must be unique throughout the IPX Internet. Valid network numbers consist of up to eight hexadecimal digits in the range &0 to &FFFFFFFD. The network number &FFFFFFFE and &FFFFFFF are reserved.

The routers use the internal network number during IPXWAN negotiation to determine which router is the master and which router is the slave. The router with the lowest internal network number becomes the slave during link establishment and information exchange.

As shown in Figure 193, the 3Com bridge/router has the highest internal network number and is designated as the master over both Novell router #1 and #2. When packets are routed between the 3Com bridge/router and the Novell routers, the network number of the bridge/router is used. Consequently, the network numbers on port 3 and port 4 of the Novell routers do not need to be assigned.

If you assign network numbers to port 3 and port 4 of the Novell routers, the 3Com bridge/router negotiates the network numbers, and the network number of the master is used. For example in Figure 193, on port 3 of the 3Com bridge/router, network number &123 is used; on port 4 of the 3Com bridge/router, network number &131 is used during packet transmission.

3 For network management purposes, assign a symbolic name to each router.

The router uses this name during IPXWAN negotiation to build NRIP/SAP Information Request/Response packets. The router name must be unique throughout the IPX Internet and can be up to 48 characters in length.

To assign a symbolic name, use:

```
SETDefault -IPX RouterName = "<string>"
```

Because the IPX router does not provide a service, the router name is not advertised in SAP updates, which substantially reduces the network traffic in a large network configuration.

4 Determine whether to use periodic or nonperiodic (incremental) NRIP/SAP update modes on your LAN or WAN ports.



All participating routers and servers must use the same update mode to avoid stale NRIP and SAP entries and loss of network connectivity.

When used in a stable and reliable network, nonperiodic updates can eliminate the constant and expensive network traffic of IPX NRIP and SAP updates on all media, except at initialization time. After initialization, updates also are sent incrementally when changes occur.

For a LAN, use periodic updates. If two bridge/routers are connected over a WAN, use nonperiodic updates. Use periodic updates on the WAN only when mixing 3Com and non-3Com routers on the same WAN link.

To enable nonperiodic updates, use:

```
SETDefault !<port> -NRIP CONTrol = NoPEriodic
SETDefault !<port> -SAP CONTrol = NoPEriodic
```

As shown in Figure 193, Novell router #1 (port 3) and the bridge/router (port 3) are configured for nonperiodic NRIP and SAP updates. The IPX router sends out NRIP and SAP updates immediately after a LAN or WAN path comes up, which completes NRIP and SAP updates more quickly.

Set the -NRIP and -SAP CONTROL parameter to PERiodic on networks in which frequent topology changes occur.

- 5 Enable the IPXWAN protocol on the specified port of each 3Com router using:

```
SETDefault !<port> -IPX CONTROL = IpxWan
```

Configuring for NLSP

The NLSP provides a hierarchical structure for large IPX routing environments. NLSP uses a link-state routing algorithm that provides faster network convergence with reduced network resource overhead (bandwidth and CPU cycles) than other routing algorithms, for example, NRIP and SAP, which use a distance vector algorithm.

NLSP runs over all networking media, including LANs (Ethernet, token ring, and FDDI), and WAN/MAN (X.25, Frame Relay, ATM, SMDS, and PPP links).

Prerequisites

Before beginning this procedure, perform the following steps:

- Configure an internal network number (-IPX InternalNET parameter) to the router.
- Configure IPX network numbers on all the LAN and WAN ports.
- Enable IPX routing on those ports.
- If there are multiple logical networks on a port, make sure the primary network is configured the same for all routers on the LAN. NLSP routes communicate with each other using the primary network only.

Procedure

To configure NLSP, follow these steps:

- 1 Determine and assign the area address for the router using:

```
ADD -NLSP AreaAddress <net> <mask>
```

NLSP uses a portion of the 32-bit IPX network number to identify an area. The AreaAddress parameter is used to describe the value and length of the area number. The area address is a pair of 32-bit integers expressed in hexadecimal format. The first set of numbers identified as <net> describes the value of the area number, while the second set identified as <mask> determines the length of the area address, or number of bits in the IPX network number field that are used to identify the area.

The mask is a number of leading 1 bits, followed by 0 bits. The leading 1 bits must be contiguous. Similar to the concept of IP subnet masks, the number of leading 1 bits in the mask determines the number of leading bits in the <net> field, which is considered to be the area number instead of the network number. Any bit position identified by a 0 in the mask is considered to be the network number. The following example shows the syntax of the area and mask:

```
ADD -NLSP AreaAddress 12345600 FFFFFFF00
```

The mask of FFFFFFF00 indicates that the first 6 characters (24 bits) in the <net> field are considered to be the area number; the last two characters (8 bits) are used to identify a network within that area. The network number is defined using the -IPX NETnumber parameter.

All network numbers assigned to routers within an area must fall within a configured area prefix. In this example, any router within the area identified as

AreaAddress 12345600 FFFFFFF00 must be assigned network numbers beginning with the prefix 123456XX. The valid range for network numbers within this area is 12345600–123456FF.

An area address must meet the following requirements:

- A mask is required, identifying a range of networks residing within the area.
- For example, all network numbers in the range 12345600 to 123456FF reside within the area 12345600 to FFFFFFF00. It is not necessary that all of the area network numbers are addressed and operational.
- All network numbers within the area must fall within the address range.
- With an area address of 12345600 FFFFFFF00, all IPX networks must begin with 123456XX. The area address 00000000 00000000 is the default and this area address includes all IPX network numbers.

2 Determine which interfaces to enable for NLSP.

NLSP routing should be enabled on all ports, including ports that have no NLSP routers connected to them. When NLSP is enabled on a port, and if there are other NRIP and SAP routers on the same port, NLSP automatically imports the NRIP and SAP information into the NLSP domain. NLSP automatically exports NLSP learned information to NRIP and SAP routers. The importing and exporting of information allows smooth operation between NLSP and non-NLSP routers.

If NLSP is disabled on the port, the import and export of NRIP and SAP routing information does not occur and causes network segmentation.

If you set the -NRIP and -SAP CONTROL parameters to Auto, the NLSP router determines if NRIP and SAP need to be enabled on the port. When the router detects a non-NLSP router or file server, it enables both NRIP and SAP to communicate with them; otherwise, it disables NRIP and SAP to conserve bandwidth.

If NLSP is disabled, the Auto setting for -NRIP and -SAP CONTROL means that both protocols are talking and listening. NRIP and SAP updates are continuously sent out. To disable these updates, use the NoTalk and/or NoListen values to override the Auto value.

By disabling NRIP and SAP, you conserve network bandwidth which is useful over Frame Relay or PPP lines. If Auto is selected for NRIP and SAP and all routers on the network support NLSP, and NLSP is enabled, the RIP/SAP traffic will automatically disappear from the network (except where file servers are present).

Enable the NLSP protocol on the specified port of each 3Com router using:

```
SETDefault !<port> -NLSP CONTROL = Enable
```

3 Display the configuration information for all ports by entering:

```
SHow -NLSP CONFIguration
```

4 Display the NLSP adjacencies by entering:

```
SHow -NLSP ADJAcencies
```

Verifying the Configuration

This section explains how to verify the status of networks that are reachable from the router and how to get statistics from the router and from other networks and stations.

Before you use the router for interconnecting networks, verify the following items on your network:

- 1 Check the state of the current configuration by entering:

SHoW -IPX DIAGnoStics

The diagnostics command will display any configuration errors that have occurred.

- 2 Check the state of the NRIP and SAP Services by entering:

SHoW -NRIP CONTroL

SHoW -SAP CONTroL

The control parameter for both of these services should be set to Talk and Listen to enable dynamic route learning.

- 3 Check the state of all networks assigned to the ports of a router by entering:

SHoW -IPX NETnumber

SHoW -IPX CONTroL

The first command displays the network numbers assigned to each port on this router and the state that each network is in. Each network should be in the UP state. If a network is in the DOWN state, check that the -IPX CONTroL parameter is enabled. If the network is in the DISABLE state, make sure that all PORT and PATH parameters are configured appropriately. The second command allows you to verify if routing is enabled on the ports.

- 4 Verify that the router can access the networks it was configured to access by entering:

SHoW -IPX AllRoutes Long

This command displays all known routes (dynamic, static, and default, if configured), hop counts, and cost in the IPX Routing Table. Adding "Long" to the command also displays gateway information.

- 5 Verify that the router can learn and exchange service information from servers on the directly connected networks and other routers, by entering:

SHoW -IPX AllServers Long

The router displays a server table. For more information on the contents of the server table, see "Learning Routes and Service Information" later in this chapter. Adding "Long" to the command also displays gateway information.

- 6 Display the configuration information for all paths by entering:

SHoW -PATH CONFIguration

Check that the configuration information is correct for all paths.

- 7 Display the configuration information for all ports by entering:

SHoW -PORT CONFIguration

Check that the configuration information is correct for all ports.

- 8 Verify the setting of the -PORT ProtMacAddrFmt parameter using:

SHoW !<port> -PORT ProtMacAddrFmt

If you did not configure the ProtMacAddrFmt parameter, the software automatically selects DefaultIPX, and based on the port media type, automatically selects either DefaultIPX(NC) for the SuperStack II NETBuilder bridge/router LAN and high-speed serial (HSS) ports or DefaultIPX(C) for Ethernet, FDDI, and HSS ports.

For more information about this parameter, see the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

- 9 Display the current IPX configuration parameters by entering:

```
SHoW -IPX CONFIguration
```

Check that the configuration information is correct.

- 10 Determine connectivity to an IPX node on the network using:

```
NetwarePING &<network>%<host> [timeout (1-300 seconds)]
```

- 11 Display the NLSP, NRIP, and SAP Services and verify the configuration information by entering:

```
SHoW -NLSP CONFIguration
```

```
SHoW -NRIP CONFIguration
```

```
SHoW -SAP CONFIguration
```

- 12 Make a connection from a workstation on one attached network to a file server on another network to see if packets can be routed across the router.

- 13 Obtain configuration information from a NetWare server using:

```
NetwareView &<network>%<host> [timeout (1-300 seconds)]
```

- 14 Obtain the status of the router by entering:

```
SHoW -IPX DIAGnostics
```

Getting Statistics

To view statistics, enter:

```
SHoW -SYS STATistics -IPX
```

```
SHoW -SYS STATistics -NRIP
```

```
SHoW -SYS STATistics -SAP
```

```
SHoW -SYS STATistics -NLSP
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information on these parameters, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*. For information on interpreting the statistics displays, see the Statistics Displays appendix.

Troubleshooting the Configuration

If you are unable to make connections to other networks after setting up the router, review the following troubleshooting procedure. This procedure can help correct problems in making single-hop (involving one router) and multiple-hop (involving more than one router) connections. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier for assistance.

To troubleshoot the configuration, follow these steps:

- 1 If you are experiencing problems because of configuration errors, examine the service diagnostics information using:

```
SHoW !<port> -IPX DIAGnostics
```

The diagnostics command displays troubleshooting information about IPX routing and gives suggestions for corrective actions. The troubleshooting information consists of global diagnostic messages, port specific diagnostic messages, NRIP diagnostic messages, and SAP diagnostic messages.

The following display appears:

```
-----IPX Diagnostic Information-----
```

```

No global diagnostic information available.
-----Port 1-----
This port seems to be normal.
-----Port 2-----
Network &00000300 conflicts with &DDDDD200 on node 080002A078DB.
-----Port 3-----
IPX Routing is not enabled.Please configure IPX CONTROL parameter.
-----Port 4-----
IPX Routing is not enabled.Please configure IPX CONTROL parameter.

```

In this example, the network assigned to port 2 is shown as &00000300, but node 2 at 080002A078DB thinks that the network should be &DDDDD200.

- 2 Make sure all cables are properly connected and that the router is properly installed.

For installation instructions, see the installation guide provided with your bridge/router.

- 3 Verify that routing is enabled by entering:

```
SHoW -IPX CONTroL
```

The router displays the current values for the CONTROL parameter for each port. If the values are set to ROUTe, no action is necessary. If the values are set to NoROUTe, to enable the IPX router use:

```
SETDefault !<port> -IPX CONTROL = ROUTe
```

- 4 Check the network number and status by entering:

```
SHoW -IPX NETnumber
```

Look at the status of the networks: each configured network should be in the UP state. If it is in the DOWN state, check to make sure that all PORT and PATH parameters are configured correctly. If the port is in the DISABLED state, make sure IPX routing is enabled for the port.

Look at the current network configuration: if no network is configured on the specific port, to add a network number to that port use:

```
SETDefault !<port> -IPX NETnumber = &<number>(0-FFFFFFFD) [Ethernet | Ieee
| Llc | Snap | X25 | PPP | Frame | SMDS | ATM]
```

Make sure that you assign the network number to the correct port. Network numbers consist of eight hexadecimal digits. For example, to assign network number 4321 to port 2 on the router, enter:

```
SETDefault !2 -IPX NETnumber = &4321
```

If this is an Ethernet port, all IPX packets sent from this port will be encapsulated with the IEEE header format, because IEEE is the default format and no format is specified in the command. Make sure that the header type configured matches that of the NetWare servers and clients.



The software detects the media type and sets the header format correctly. You do not need to specify the header format type in the NETnumber parameter for a WAN. Each LAN must be configured if the default is not appropriate.

To detect a mismatch of encapsulation type or network, enter:

```
SHoW -IPX DIAGnostics
```

- 5 Verify that dynamic learning and NRIP updates are enabled on the port by entering:

SHoW -NRIP CONTroL

The router displays the current values for the NRIP CONTroL parameter. If dynamic learning and NRIP updates are disabled on the port, enable it using:

```
SETDefault !<port> -NRIP CONTroL = (Talk, Listen, PEriodic)
SETDefault !<port> -SAP CONTROL = (Talk, Listen, PEriodic)
```

- 6 Verify that the network you are trying to reach is in the IPX Routing Table by entering:

SHoW -IPX AllRoutes

The IPX router displays the routing table entries. From the table entry, you can determine which path is being used. Examine the entries to make sure a route in the table is taking the appropriate path. You can also specify a network number using the SHoW -IPX AllRoutes <NETnumber> syntax to verify single route reachability.

If the entry in the table has a hop number of 16, the network is unreachable at the present time. Wait several minutes and use the SHoW -IPX AllRoutes <NETnumber> syntax again. Optionally, you can use the FLush -IPX AllRoutes command to remove dynamically learned routes and services. After flushing the table, wait a few minutes before reentering the SHoW -IPX AllRoutes command.

- 7 Verify that the server you are trying to reach is in the IPX Server Table by entering:

SHoW -IPX AllServers

The IPX router displays all known servers in the IPX Server Table, including server addresses, server names, and the number of hops involved. Make sure the server name to which you are trying to connect is in the table.

You can also specify a server name using the SHoW -IPX AllServers " <string>" syntax to verify single server reachability.

- 8 If you are experiencing connectivity problems due to routing and service tables that are not synchronized between IPX routers on your internetwork, flush the routing and service table entries by entering:

FLush -IPX AllRoutes**FLush -IPX AllServers**

These commands remove all dynamically learned entries from the routing table and all entries from the server table, and then rebuild these tables.

- 9 Display statistics for the IPX Service by entering:

SHoW -SYS STATistics -IPX

For information on interpreting statistics displays, see the Statistics Displays appendix.

- 10 Display statistics for the NRIP and SAP Services by entering:

SHoW -SYS STATistics -NRIP**SHoW -SYS STATistics -SAP**

Customizing the IPX Router

This section provides additional procedures you can use to configure your IPX router.

Controlling NRIP and SAP Advertisements

The -NRIP and -SAP CONTROL parameters determine how the router sends the routing table information to the network. For information on periodic and nonperiodic NRIP and SAP updates, see “Controlling NRIP and SAP Updates” later in this chapter.

You only need to specify values that differ from the default values.

Enabling and Disabling Dynamic Learning and NRIP Updates

The router maintains a routing table of all networks it can reach. The router adds routes to the routing table automatically from its neighbors’ route advertisements unless you disable dynamic learning. If you do not disable dynamic learning, the router eventually will learn all of the networks it can reach.

Enable dynamic learning for a given port using:

```
SETDefault !<port> -NRIP CONTROL = Listen
```

For ports that connect non-broadcast multiaccess (NBMA) networks, you can enable dynamic neighbor learning using:

```
SETDefault !<port> -NRIP CONTROL = DynamicNbr
```

You may want to disable dynamic learning if you are configuring static routing on a port and want to eliminate traffic associated with the route advertisements. Disabling dynamic learning frees bandwidth on slow serial data links and is especially cost-effective on an X.25 or Frame Relay interface where packet charges are enforced. Disable dynamic learning using:

```
SETDefault !<port> -NRIP CONTROL = (NoListen, NoTalk)
```

For ports that connect NBMA networks, you can disable dynamic neighbor learning using:

```
SETDefault !<port> -NRIP CONTROL = NoDynamicNbr
```

The effect of setting the Listen | NoListen value for the -NRIP CONTROL parameter depends on the setting of the ROute | NoROute value for the -IPX CONTROL parameter. If the -IPX CONTROL parameter is set to NoROute, dynamic learning is disabled and the NRIP update is also disabled.



If you disable dynamic learning, you must add a static route for each network to which you want to connect. For more information, see “Adding and Deleting Static Routes” later in this chapter.

For a description of additional -NRIP CONTROL parameter values, see the NRIP Service Parameters chapter in *Reference for Enterprise OS Software*. For a discussion of split horizon, see “Solving the Slow Convergence Problem with Poison Reverse” later in this chapter.

Enabling Triggered NRIP Updates

Setting the -NRIP CONTROL parameter to Trigger causes the router to send an update packet when the network topology reflected in its routing table changes. The advantage is that the network immediately knows a potentially better route to a particular network. Setting the -NRIP CONTROL parameter to NoTrigger reduces the amount of data packets broadcast over the network during topology changes, and normal update packets will be sent only at the time interval specified by the UpdateTime parameter.

Enable the trigger feature for a given port using:

```
SETDefault !<port> -NRIP CONTrol = Trigger
```

Using Poison Reverse or No Poison Reverse

The poison reverse and no poison reverse implementations are described in detail in “Solving the Slow Convergence Problem with Poison Reverse” later in this chapter.

To enable the poison reverse feature for a given port, use:

```
SETDefault !<port> -NRIP CONTrol = POison
```

To disable the poison reverse feature for a given port, use:

```
SETDefault !<port> -NRIP CONTrol = NoPOison
```

Another way to solve the slow convergence problem is to run NLSP instead of NRIP and SAP.

Controlling NRIP and SAP Updates

In a stable and reliable network in which topology changes are infrequent, you can eliminate most of the traffic of NRIP and SAP updates by using the NoPEriodic values of the -NRIP and -SAP CONTrol parameters. You can select these values using:

```
SETDefault !<port> -NRIP CONTrol = NoPEriodic  
SETDefault !<port> -SAP CONTrol = NoPEriodic
```

You can always use NoPEriodic on WANs. NoPEriodic can only be used on LANs if the servers also support NoPEriodic. The default setting for a WAN is NoPEriodic, the default for a LAN is PEriodic. Because multiprotocol ATM is treated like a LAN, it also uses the PEriodic setting for its default. Do not change the setting to NoPEriodic on a LAN unless NoPEriodic is supported by the servers.

When you select these values, the IPX router shuts off periodic NRIP and SAP updates and switches to incremental updates, allowing the transmission of updates only when topology changes occur. When selecting these options, make sure that all participating routers use the same option. You can use the PEriodic and NoPEriodic settings for NRIP and SAP for all media.



If you are using the Boundary Routing system architecture, use smart filters and NoPEriodic on the WAN links to the remote sites.

If your network has frequent topology changes, NRIP and SAP updates need to occur on a periodic basis. However, setting NRIP and SAP updates to a periodic basis should only be used on the WAN when mixing 3Com and non-3Com routers on the same link. Selecting periodic updates in an all-3Com network can create severe traffic problems. If you have a mixed network on the WAN link, you can enable the periodic updates using:

```
SETDefault !<port> -NRIP CONTrol = PEriodic  
SETDefault !<port> -SAP CONTrol = PEriodic
```

When you select these values, the IPX router sends NRIP and SAP updates when topology changes occur (triggered updates) and each time the value of UpdateTime parameter expires.

You can use the Auto option for NRIP and SAP to allow the router to transition from the RIP and SAP routing protocols to NLSP as NLSP routers are configured on the network. If there is a mix of RIP, SAP, and NLSP routers on a LAN, and the NLSP routers have selected Auto, the NLSP routers will interoperate with the NRIP and SAP routers by announcing NLSP learned routes and services to the non-NLSP routers using RIP and SAP updates.

Learning of routes and services continue in the same way. When the last router on a network is configured with NLSP, all RIP and SAP traffic automatically disappears from the network unless there are file servers present. The Auto option can be overridden in the NRIP and SAP Services if you require some flexibility in the control of your network.

For conceptual information, see “Learning Routes and Service Information” later in this chapter. To eliminate NRIP and SAP updates in a Boundary Routing environment, you can use the smart filtering feature; for more information, see the Configuring Boundary Routing System Architecture chapter. Another way to solve this problem is to run NLSP instead of NRIP and SAP.

Controlling Route and Service Aging

The UpdateTime parameter controls learned route aging when dynamic learning is enabled: the router purges learned routes from its routing table if they are not readvertised by a neighbor within three times the update time interval. The UpdateTime parameter also defines, in seconds, the interval at which the router generates NRIP and SAP updates. Valid settings are integer values from 10 to 65535; for most situations, the default setting of 60 seconds is sufficient.



CAUTION: *To avoid loss of connectivity, make sure all nodes on the network are set to the same UpdateTime value. Because NetWare servers use a default of 60 seconds, make sure all nodes on the network are set to the same value especially if you have NetWare servers on your network.*

Set the update time interval using:

```
SETDefault !<port> -NRIP UpdateTime = <seconds>(10-65535)
or
SETDefault !<port> -SAP UpdateTime = <seconds>(10-65535)
```

A higher value for the UpdateTime parameter increases the time it takes for all routers on the network to converge on the same topology and allows dynamic learning to occur. A lower value reduces the time it takes for the convergence to occur, at the expense of network overhead. All routers on the same network must have the same UpdateTime value.

Another way to solve this problem is to run NLSP instead of NRIP and SAP.

Flushing Dynamic Routes and Server Table Entries

If you are experiencing connectivity problems due to routing and service tables that are not synchronized between IPX routers on your internetwork, you can flush the route and server table entries instead of waiting for them to time out or powering down each router.

To remove all entries learned dynamically from the routing table, enter:

```
FLush -IPX AllRoutes
```

To remove all entries from the server table, enter:

```
FLush -IPX AllServers
```

For more information on the FLush command, see the Commands chapter in *Reference for Enterprise OS Software*.

Flushing Dynamically Learned WAN Neighbors

IPX allows dynamic learning of neighbors from its WAN interface. If you need to change the WAN interface type, you can also flush the dynamically learned neighbors. To flush the neighbor table, use:

```
FLush !<port> -IPX ADDRESS
```

If further regular updates are not received, dynamically learned WAN neighbors can also be aged out of the neighbor table.

Built-in IPX Masks

Table 48 lists the built-in IPX masks. These predefined masks identify different types of IPX packets. To display this table, enter:

```
SHow -Filter MASK BuiltIn
```

Table 48 Built-in IPX Masks

| Built-in Mask | Use |
|---------------|--|
| IPXRIP | Matches a RIP Packet |
| SAP | Matches a SAP packet |
| FSP | Matches a Netware File Service NCP packet |
| WANBC | Matches a broadcast packet of IPX packet type 20 |
| TRACERT | Matches a 3Com-proprietary Trace packet (soc = 0x874e) |
| IPXPING | Matches an IPX Ping packet (soc = 0x9086) |
| IPXDIAG | Matches an IPX Diagnostic packet (soc 0x456) |
| NWSEC | Matches a Netware Security packet (soc = 0x457) |

User-defined IPX Masks

Table 49 lists valid IPX field mnemonics and match values. You can use these field mnemonics to specify the offset or location in an IPX packet. ALL is a valid match mnemonic for certain field categories. When ALL is specified, any value in the location is considered to match the criteria. The percent sign (%) is used to specify a hexadecimal value; otherwise, the value is considered to be decimal.

To display a list of valid locations supported for IPX, enter:

```
SHow -Filter MNEmonics
```

Table 49 IPX Built-in Mnemonics for User-defined Masks

| Field | Description | Matching Value |
|------------|--|--------------------------------------|
| DstNETwork | IPX destination network | %<hexadecimal network number>
ALL |
| SrcNETwork | IPX source network | %<hexadecimal network number>
ALL |
| NETwork | Either IPX destination or source network | %<hexadecimal network number>
ALL |

Table 49 IPX Built-in Mnemonics for User-defined Masks (continued)

| Field | Description | Matching Value |
|--------------------------------|---|---|
| DstNodeAddr | IPX destination node address | %<hexadecimal node address> |
| SrcNodeAddr | IPX source node address | %<hexadecimal node address> |
| NodeAddr | Either IPX destination or source node address | %< hexadecimal node address> |
| DstSockeT | IPX destination socket | FileServicePacket
ServerAdvPkt
RouteInfoPkt
IpxPING
IpxDIAG
IpxTraceRt
NWSecPkt
%<hexadecimal socket number> |
| SrcSockeT | IPX source socket | FileServicePacket
ServerAdvPkt
RouteInfoPkt
IpxPING
IpxDIAG
IpxTraceRt
NWSecPkt
%<hexadecimal socket number> |
| (continued) | | |
| SockeT | Either IPX destination or source socket | FileServicePacket
ServerAdvPkt
RouteInfoPkt
IpxPING
IpxDIAG
IpxTraceRt
NWSecPkt
%<hexadecimal socket number> |
| PacketLength | IPX packet length | %<hexadecimal value> |
| PacketType | IPX packet type | %<hexadecimal value> |
| TransportCtl | IPX transport control | %<hexadecimal value> |
| DATA+[%]<offset>[:[%]<length>] | Starting <offset> bytes after the end of the IPX header and <length> bytes long | %<hexadecimal value>
<" ascii string" > |



The maximum length allowed for a string-match value is 10. For a numerical value, only 1, 2, or 4 are valid lengths.

Adding and Deleting Static Routes

Routes dynamically learned are automatically purged from the routing table if they are not readvertised within a certain period of time (for details see "Controlling Route and Service Aging" earlier in this chapter).

If you want to add a route to the routing table that will not be purged from the table, eliminate route advertisements required for dynamic route learning, and

optimize the use of the available bandwidth on slow serial data links, you must add the route as a static route.

If a destination network is reachable with both a static route and a learned route, the router uses the static route unless you specify the optional Override value in the ADD ROUTe command. If a learned route of higher precedence is available, it overrides the static route.

If you want to eliminate NRIP and SAP advertisements (bandwidth protection), you can configure the bridge/router for nonperiodic updates through the -NRIP and -SAP CONTROL parameters. For more information, see “Controlling NRIP and SAP Updates” earlier in this chapter.

The IPX router ignores any dynamic updates or backup routes on the network when a static route is configured for a specific network. Static routes are recommended only where the network topology remains constant.

Prerequisites

For each router port on which you want to add static routes, you must configure a network number (see “Setting Up a Basic IPX Router” earlier in this chapter).

Procedure

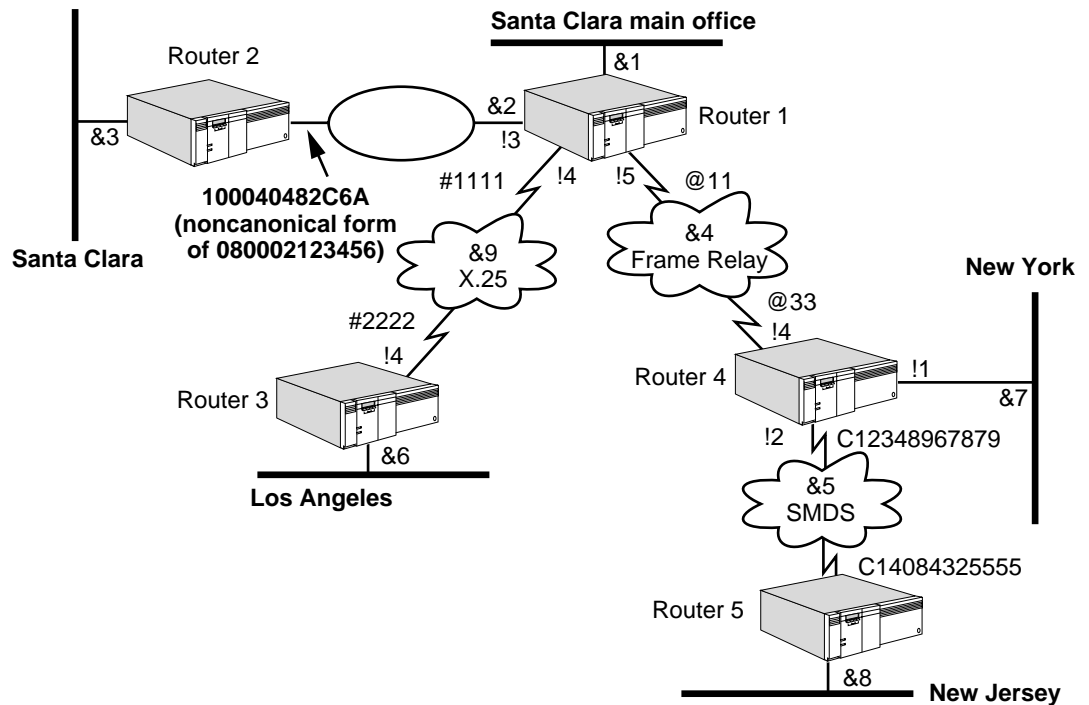
Define a static route using:

```
ADD !<port> -IPX ROUTe {&<remote network> | Default} [<network>] <media  
address> <hops> [hdrfmt]
```

Figure 194 shows router 1 (Santa Clara Office) that can reach three remote routers through different media:

- Router 2 (Santa Clara Branch) is reachable by LAN.
Whenever token ring is involved, as in this example, make sure that the ProtMacAddrFmt parameter is set to the correct address format.
- Router 3 (Los Angeles Branch) is reachable by X.25.
- Router 4 (New York Branch) is reachable by Frame Relay.
- Router 5 (New Jersey Branch) is reachable by SMDS indirectly through router 4.

Figure 194 Adding Static Routes



Steps 1 through 4 of the following procedure are performed either from a console attached to router 1 or via a Telnet connection to router 1. To communicate with router 1 via Telnet, router 1 must have an IP address that is reachable from the workstation or router console from which the Telnet connection is initiated.

On router 1, follow these steps:

- 1 Add a static route to Santa Clara (network &3) by entering:

```
ADD !3 -IPX ROUTE &3 %100040482C6A 1
```

This command specifies that network 3 is reachable through the device identified by MAC address %100040482C6A on network &2, and that the route to network &3 has a hop count of 1. The command is identical for all neighbors reachable through LAN connections (Ethernet, FDDI, or token ring) except the MAC address, which must be set appropriately depending on the -PORT ProtMacAddrFmt parameter value.

- 2 Add a static route to Los Angeles (network &6) by entering:

```
ADD !4 -IPX ROUTE &6 #2222 1
```

- 3 Add a static route to New York (network &7) by entering:

```
ADD -IPX ROUTE &7 &4 @33 1
```

- 4 Add a static route to New Jersey (network &8) by entering:

```
ADD !5 -IPX ROUTE &8 @33 2
```

The routes to networks &7 and &8 (defined in steps 3 and 4, respectively) are identical except for the destination network identifier and the hop count. This is because the "next hop" for any packet routed by router 1 to either network &7 or &8 is router 4.

- 5 If dynamic learning is disabled on router 4, you will also need to add a static route from Router 4 to network &8 by entering:

```
ADD !2 -IPX ROUTe &8 $C14084325555 1
```

To display the static routes configured far, enter:

```
SHow -IPX ROUTe
```

Because static routes do not age out, they must be removed manually. To delete a static route, use:

```
DELete -IPX ROUTe &<remote network>
```

Configuring a Static Default Route

You can configure a static default route, which is subsequently added to the routing table and propagated by NRIP or NLSP. Once a default route is specified, packets destined to unknown networks (networks not explicitly known or listed in the routing table) are routed to the default router for subsequent routing. You can configure only one default route per port.

Use this procedure to configure a default route so that unknown destination packets can be properly forwarded. For conceptual information, see “Default Routes” later in this chapter.

Procedure

To configure a default route, see Figure 195 and follow these steps:

- 1 Assign a default route on the router port to point to the default router using:

```
ADD !<port> -IPX ROUTe Default <media address> <hop>
```

Substitute the MAC address of the default router for <media address>.

For example, in Figure 195, configure the ROUTe parameter on port 3 of Router B and use the MAC address of Router A. Router B adds a static route (labeled as the default route) to its routing table and advertises the route to downstream routers (Router C and Router D). When Router B receives an unknown destination packet (for example, a client on network &600 transmits a packet destined for network &1000), Router B uses the default route, and routes the packet out port 3 to the default router, which sends the packet toward its destination.

The special network number &FFFFFFE has been reserved for the default route. If this address has already been used within an organization, it must be renumbered.

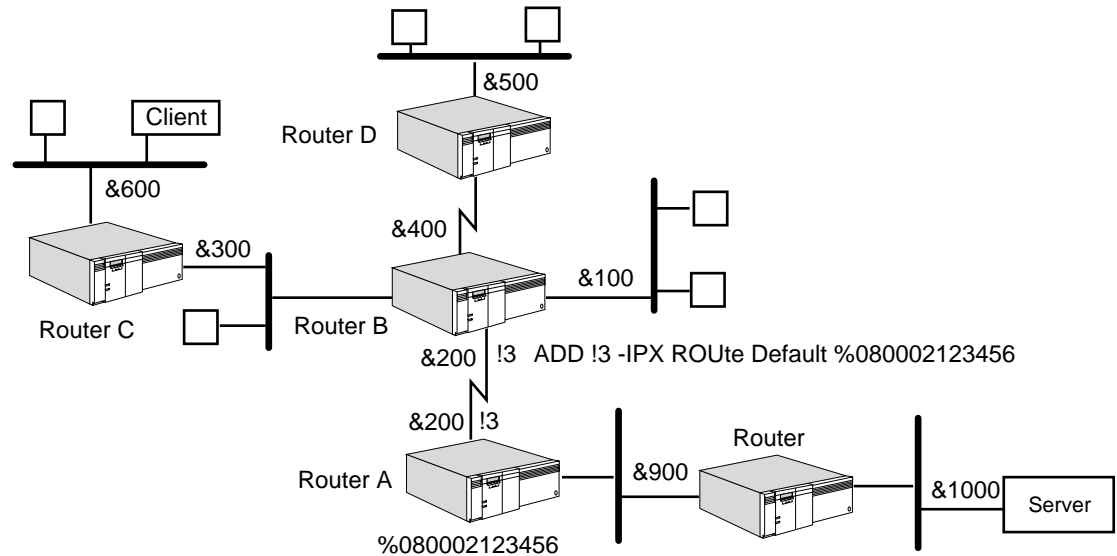


The default route implementation for NRIP is not supported in software versions prior to version 8.2. All routers must be upgraded to software version 8.2 or later so that all NRIP routers recognize &FFFFFFE as the default route and forward packets for unknown destinations toward it.

- 2 Verify that the default route has been added to the routing table of the remote router by entering:

```
SHow -IPX AllRoutes
```

The default route is the first route in the routing table and is labeled Default.

Figure 195 Configuring an IPX Default Route

Configuring a Default Metric

You can configure a default metric on a router to advertise a default route to other routers. The default metric allows default route advertisements to be sent in RIP updates. Other routers, receiving such advertisements, send all unknown destination packets to this router. Without default route advertisements, unknown destination packets are dropped before they can reach this router.

To configure a default metric on a router and enable default route advertisement, see to Figure 196 and use:

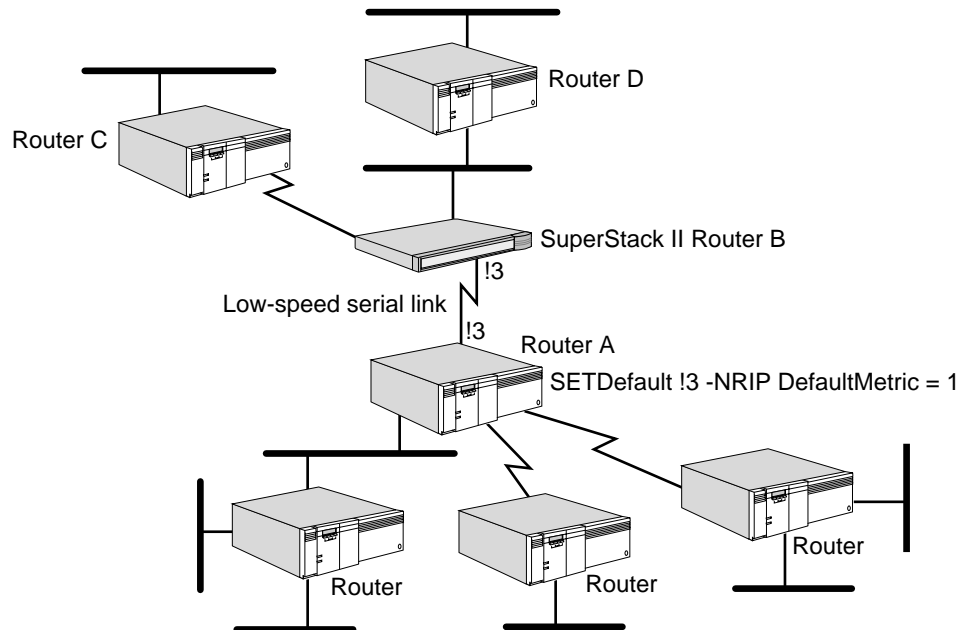
```
SETDefault !<port> -NRIP DefaultMetric = <hops(1-15)> [<ticks>]
```

For <hops>, select a value between 1 and 15 to enable advertisement of this route within the corresponding hop count. For example, on port 3 of router A, configure a default metric of 1.

Optionally, for <ticks>, you can select a value between 1 and 65535. If there is more than one route to the same destination, the router uses the one that has the lowest tick value.

For example, router A advertises the default route over port 3. When router B receives the advertisement, it adds a static default route to its routing table and propagates the metric to the other downstream routers (routers C and D). When the downstream routers receive unknown destination packets, they route them for router B, which uses the default route in its routing table to route the packet over port 3.

By using the default metric, the routing tables of the remote routers (routers C and D in the figure) can be reduced in size; the routing tables of routers C and D do not need to contain routes to router A or have knowledge of other networks that are attached to router A.

Figure 196 Configuring an IPX Default Metric

Adding and Deleting Static Servers

The IPX Service allows you to enter static servers into the SAP information table. The static server will not be aged out, so it will not be purged in the aging-out time frame. The IPX router dynamically updates the server once a static server is configured. Specifying a static server is recommended only when the network topology remains constant.

Define a static route using:

```
ADD -IPX SERver <sname> <type> <snet>%<shost>:<sskt> <hops>
```

Where <sname> is the name of the static server being configured, <type> is the type of service, <snet>%<shost>:<sskt> is the server address and <hops> is the hop count away from this IPX router. For more information, see "SERver" in the IPX Service Parameters chapter in *Reference for Enterprise OS Software*.

Configuring Neighbor Policy

When you enable route advertisements to neighbors by setting the -NRIP and -SAP PolicyControl parameters to AdvToNbr, broadcast NRIP and SAP updates are automatically disabled, and only those neighbors specified with the RcvFromNbr attribute receive unicast NRIP and SAP updates. The NRIP and SAP Services can maintain a different neighbors list.

There are two reasons to configure neighbors:

- To use the next-hop split horizon scheme on a neighbor basis, as described in "How the IPX Router Works" later in this chapter.
- To control routing domains for security (advertise routes only to specified neighbors).

To configure neighbors, follow these steps:

- 1 Enable AdvToNeighbor using:

```
SETDefault !<port> -NRIP PolicyControl = AdvToNbr
```


- 2 If dynamic neighbor learning is enabled using the CONTrol parameter, and if the port is configured for Frame Relay or X.25, then this step can be skipped. Specify all of the neighbors to which NRIP updates are to be sent using:

```
ADD !<port> -NRIP AdvToNeighbor <network>%<mac address> [...]
```

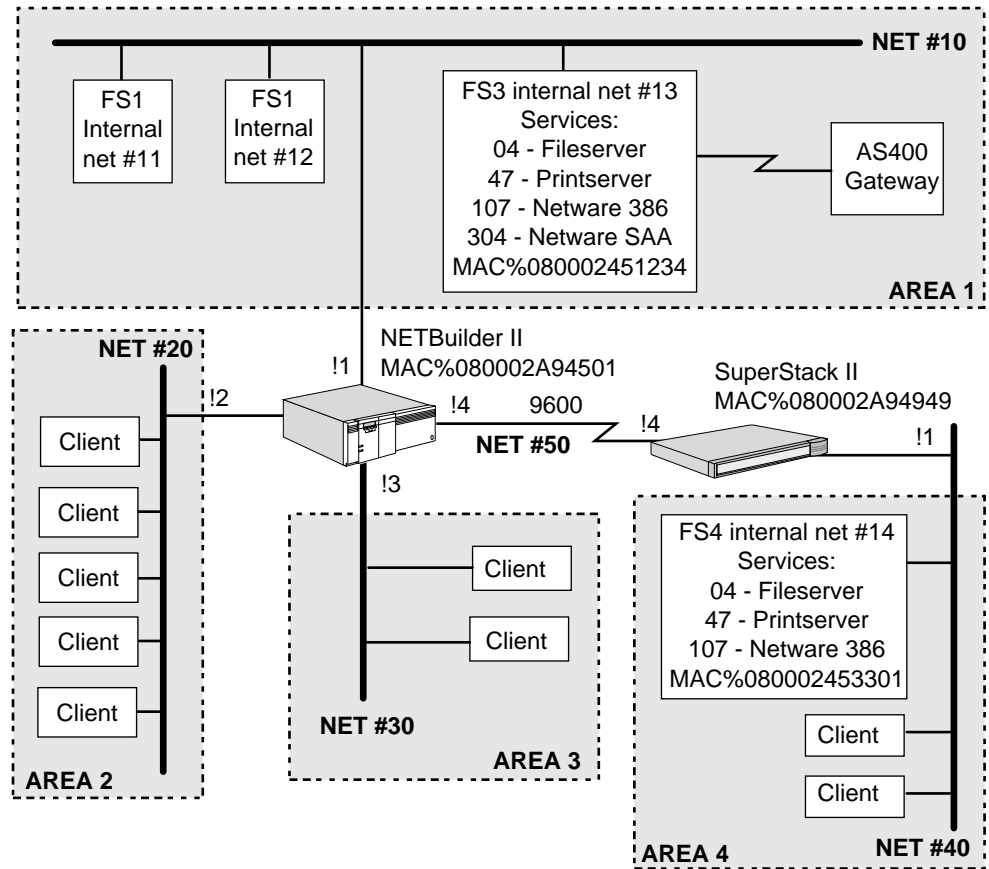
If the physical connection is made with a 3Com bridge/router that has an HSS module installed, use the MAC address of the HSS module interface connecting the neighbor to the network.

Writing NRIP and SAP Policies for IPX

- 3 Using NRIP and SAP policies can provide security, reduce route and service table sizes on file servers and bridge/routers, and help reduce excessive traffic across WAN links. The NRIP and SAP Services can maintain different policies. The policies consist of lists of network numbers, service names, and SAP types. Lists can be created as *normal lists* or *inverse lists*. Normal lists list every network number, server name, and service type that is included in a policy. Inverse lists list every network number, server name, and service type that is excluded from a policy. For background information on policies, see "Route, Service, and Neighbor Policies" later in this chapter. For a listing of Novell service advertising type descriptions, see Table 50.

Figure 197 shows a NETBuilder II bridge/router and a SuperStack II NETBuilder bridge/router serving different networks in which IPX is being used for NetWare environments. See the figure in the examples that follow.

Figure 197 Using RIP and SAP Policies in IPX Environments



NETBuilder II Examples

You can configure the NRIP and SAP policies in different ways on the NETBuilder II bridge/router, as shown in the following examples.

Example 1 Because area 2 and area 3 do not need to access the file server in area 4, and clients in area 4 do not need to communicate with clients in area 2 or area 3, their routes do not need to be broadcast across the 9600 baud link. In this situation, you could set up a route policy on the NETBuilder II bridge/router to keep traffic off the 9600 baud link by entering:

```
ADD !4 -NRIP AdvertisePolicy ~&20-30
SETDefault !4 -NRIP PolicyControl = AdvPolicy
```

The result of these commands, is that all networks except 20 and 30 are advertised to area 4.

Example 2 If no remote segments need to access the two file servers in area 1, you could write a route policy to keep the NETBuilder II bridge/router from receiving the internal IPX net number of the file server. To write such a route policy, enter:

```
ADD !1 -NRIP ReceivePolicy ~&11-12
SETDefault !1 -NRIP PolicyControl = RcvPolicy
```

Example 3 If you want other segments to have access to area 1, but not to all the services available on all file servers in area 1, you could set a SAP policy by following these steps:

1 Configure the advertise policy using:

```
ADD !<port> -SAP AdvertisePolicy
```

If you want to advertise only file server 3, you can enter the address used in the policy in three different ways:

- By specifying the internal IPX net, server host address, and service type.

For example, to set the advertise policy in this way, enter:

```
ADD !4 -SAP AdvertisePolicy &0000013:%000000000001:04
```

In this example, :04 indicates the service type in Figure 197, for example, file server.

The server address is not the 48-bit MAC address of the host on which the service is located. NetWare servers usually advertise themselves with address 000000000001. To determine the address of the server, enter:

```
SHow -IPX AllServers Long
```

If your NetWare servers advertise themselves with address 000000000001, specifying this address filters all servers on the network. To filter a server individually, specify it by server name.

- By specifying the actual file server name and service type.

For example:

```
ADD !4 -SAP AdvertisePolicy "FS3":04
```

You can also advertise all services from FS3 by using the asterisk character as a wildcard, as shown in the following example:

```
ADD !4 -SAP AdvertisePolicy "FS3":*
```

- By specifying the policy number command.

For example, if FS3 is using NetWare 2.X, in which no internal IPX network numbers are used, enter the policy number command specifying the IPX network number, server MAC address, and service type as follows:

```
ADD !1 -SAP AdvertisePolicy &0000010:%000000000001:04
```

Instead of entering:

```
ADD !1 -SAP AdvertisePolicy &0000013:%080002451234:04
```

The result of these commands is that only file server 3 is advertised to area 4. No other file servers are advertised to area 4.

2 Set the policy control to enable using:

```
SETDefault !<port> -SAP PolicyControl = AdvPolicy
```

3 To view addresses on the file servers, enter:

```
SHow -IPX AllServers Long
```

Example 4 If you need to access some, but not all services on all networks, then a SAP policy can be used to control the specific services that are available. The SAP policy on the NETBuilder II bridge/router could advertise type 4 file services and AS400 gateway (type 304) but not print service type 47 or NetWare 386 (type 107).

To keep SAP broadcasts type 47 from being received on port 1 of the NETBuilder II bridge/router (no printer SAP type 47 or NetWare 386 (type 107) from server FS3 but still receiving the file service type 4 and the AS400 gateway type 304), set the policy by entering:

```
ADD !1 -SAP ReceivePolicy ~"FS3":47
ADD !1 -SAP ReceivePolicy ~"FS3":107
SETDefault !1 -SAP PolicyControl = RcvPolicy
```

Example 5 If area 2 and area 3 need printer services available, but you do not want them advertised out to area 4, enter:

```
ADD !4 -SAP AdvertisePolicy ~"FS3":47
SETDefault !4 -SAP PolicyControl = AdvPolicy
```

SuperStack II Examples

You can configure the NRIP and SAP policies in different ways on the SuperStack II bridge/router, as shown in the following examples.

Example 1 You can define a route policy on the SuperStack II bridge/router to keep unnecessary packets off the local area network. To configure this route policy, enter:

```
ADD !1 -NRIP AdvertisePolicy ~&50
SETDefault !1 -NRIP PolicyControl = AdvPolicy
```

In this situation, the net number for the WAN link does not need to be broadcast out on the local area network.

Example 2 You can add a SAP policy for restricting the advertisement of other services (the print server and the NetWare 386) available on FS4 file services by entering:

```
ADD !1 -SAP ReceivePolicy "FS4":04
SETDefault !1 -SAP PolicyControl = RcvPolicy
```

In this situation, the receive policy keeps the SuperStack II bridge/router server from even storing the other services in its SAP table. The SuperStack II bridge/router will not respond to SAP request for service type 47 (printer) or any service on FS4 except type 04 file services.

Example 3 If clients in area 1, area 2, and area 3 do not need access to area 4, you can set a policy for this situation by entering:

```
ADD !4 -NRIP AdvertisePolicy ~&40,~&14
SETDefault !4 -NRIP PolicyControl = AdvPolicy
```

Note that in this example, &14 is the internal IPX number for FS4.

Configuring Other Policy Settings

You can configure other settings for NRIP and SAP policies, including setting up lists of IPX neighbors (next hop routers, or file servers broadcasting NRIP and SAP packets). These lists are defined by the MAC address and are used to determine who the router should accept from or advertise to the NRIP and SAP information.

These policy settings are configured using:

```
ADD !<port> -NRIP or -SAP AdvToNeighbor
ADD !<port> -NRIP or -SAP RcvFromNeighbor
```

These syntaxes are used in the same way that the following syntaxes are used in this section:

```
ADD !<port> -NRIP or -SAP AdvertisePolicy
```

ADD !<port> -NRIP or -SAP ReceivePolicy



The AdvToNeighbor parameter cannot accept the ~ (inverse) policy.

One example of where to apply a neighbor policy as shown in Figure 197 is to have the bridge/router receive NRIP broadcasts from only properly configured file servers to protect itself from servers that may send conflicting NRIP and SAP broadcasts. For example, to receive NRIP and SAP information only from FS3, enter the following commands on the NETBuilder II system:

```
ADD !1 -NRIP RcvFromNeighbor %080002451234
SETDefault !1 -NRIP PolicyControl = RcvFromNbr
```

You can use the PolicyOverride setting to override the configured policies on locally connected routers (not to be used across serial links) when the router issues responses to specific NRIP or SAP requests. Use the PolicyOverride setting when one side of a router is all clients that do not need to see NRIP and SAP broadcasts, but the 3Com bridge/router still needs to respond to RIP and SAP requests. In this case, AdvertisePolicy is set for both NRIP and SAP, and PolicyControl is configured with the AdvPolicy RcvPolicy, and PolicyOverride settings.

In this situation, on the 3Com bridge/router, leave the route list blank for the AdvertisePolicy parameter, and enable PolicyControl for both route and services being advertised. No NRIP or SAP broadcasts are advertised on port 2, but with policy override enabled, the 3Com bridge/router can still respond to the client's specific request for connections to the file servers. To set this configuration, enter:

```
SETDefault !2 -NRIP PolicyControl = (AdvPolicy, RcvPolicy, PolicyOverride)
```

Configuring IPX Spoofing over a DOD Link

To help you better control IPX traffic over DOD lines, software version 9.1 and later can spoof NetWare 3.0 and 4.0 NetWare Core Protocols (NCP) KeepAliveRequest and Sequenced Packet Exchange 1 (SPX1) keepalive packets to reduce the time a DOD line is kept in the up state. Spoofing these packets makes more efficient use of the DOD WAN links.

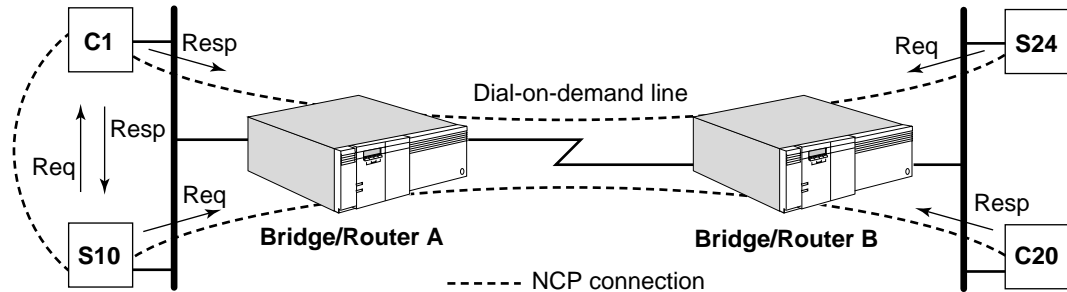
NCP Spoofing over a DOD Link

An NCP connection between a NetWare client and a server is maintained through an exchange of KeepAliveRequest and Response packets between the two. For each client connection, the server maintains information about the connection. For example, any connection that is idle or no longer used is terminated by the server, and all resources that were allocated to that connection can be reused. The server determines that a connection is no longer needed as described in the following paragraphs.

The server sends a KeepAliveRequest packet to the client to see if it is still attached to the server after a time interval has elapsed since the server last sent a NCP request or received a data transfer from the client. For example, in Figure 198,

server S10 sends KeepAliveRequest packets to clients C1 and C20; server S24 sends KeepAliveRequest packets to client C1.

Figure 198 NCP KeepAliveRequest Packet Exchange



If the server receives the KeepAliveResponse packet from the client within a certain time interval, then the client is still considered to be a client and its connection to the server is maintained. For example, in Figure 198, C1 and C20 respond to the KeepAliveRequest packets from S10 by sending KeepAliveResponse packets if they are still active; C1 also responds to the KeepAliveRequest packet from S24.

If the server does not receive a KeepAliveResponse packet from the client within a certain time interval, then the server continues to resend the KeepAliveRequest at regular time intervals until it either receives a response or it has exhausted its KeepAliveRequest retry counts. In the latter case, the client is considered to be no longer a client and its connection is terminated.

On NetWare 3.0 and 4.0 servers, the number of KeepAliveRequest retries, time interval (delay) before sending the first request, and the time interval between retries are all user-configurable parameters.

NCP Keep Alive Mechanism

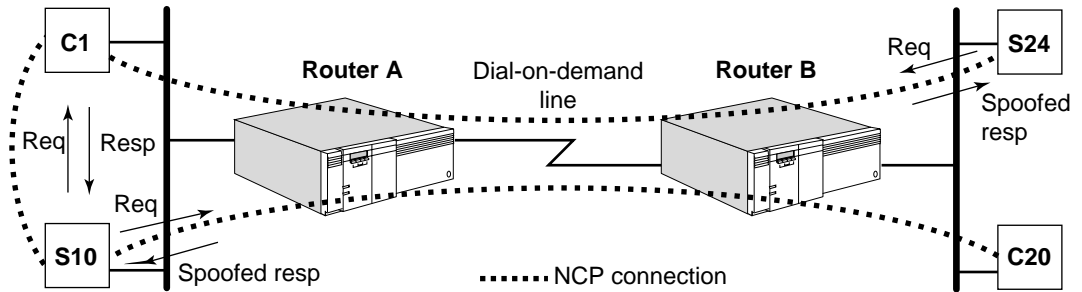
In a LAN environment and on non-DOD WAN lines where the path is always up, the keep alive mechanism operates properly. On DOD WAN lines, the function of DOD is to bring the path down and, in the absence of any other traffic over this path, keep the path down to reduce phone charges. In contrast, the sending of KeepAliveRequest packets by the server to query its client brings the path up (and down) constantly, and may also prevent a DOD path from idling to a down state.

To resolve these problems, spoofing of the NetWare 3.0 and 4.0 NCP keep alive packets has been implemented. First available in software version 8.0, spoofing is a mechanism that allows the bridge/router to respond to an incoming KeepAliveRequest packet that is to be routed over a DOD line, by sending a KeepAliveResponse packet to the originating server of the request on behalf of the intended client. The bridge/router with spoofing software spoofs only when the DOD path is down to prevent the DOD path from constantly coming up and going down due to the transmission of KeepAliveRequest packets from the server. When the DOD path is up, the bridge/router routes the KeepAliveRequest and Response packets as expected in the normal NCP connection process.

For example, in Figure 199, a quiet NCP connection over an idle DOD path exists between C1 and S24, and between C20 and S10. With spoofing, router A

responds to the request from S10 to C20 for the client. The request packet is intercepted and processed by the IPX software, replied to, and discarded without being transmitted over the DOD line. The discarded packet does not trigger the raising of the DOD path as a normal routed packet does. Similarly, Router B spoofs the request from S24 to C1. With server S10 and client C1, the normal NCP connection process occurs; the bridge/router performs spoofing only across a DOD link, not on a LAN.

Figure 199 Spoofing of KeepAliveRequest Packets over DOD Paths



The maximum number of spoofed client-server connections that are handled by the router is not limited.

In the NCP keep alive mechanism, the clients are totally passive; it is up to the server to maintain the connections based solely on the responses to its requests. This characteristic allows the spoofing software to spoof only on the server side (spoofer response) without having to worry about the client side (spoofer request).

Supported Configurations

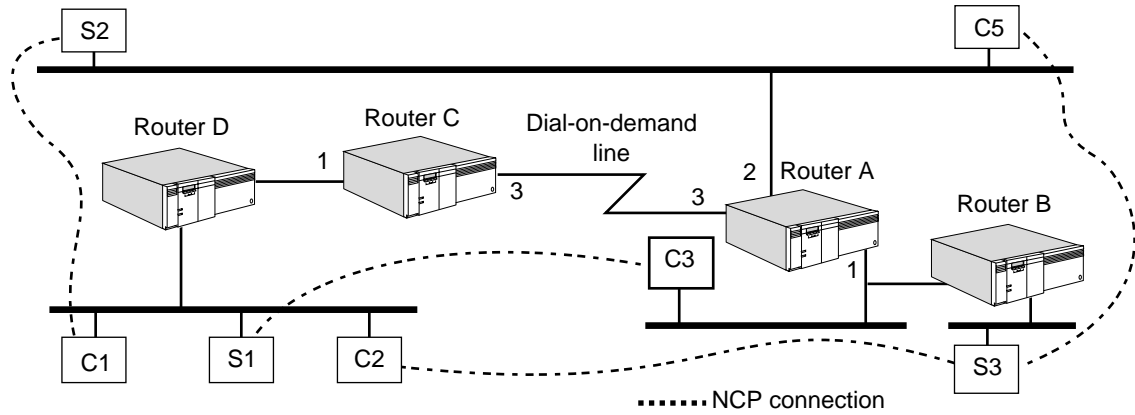
The NCP spoofing feature can be used with DOD in the following IPX network configurations:

- Router-to-router configuration
- Boundary Routing configuration

In a router-to-router configuration, NCP spoofing can be used to support symmetrical two-way client-server access. For example, clients on either side of a DOD line can access servers on either side of the line with NCP spoofing of the server's keep alive requests. For example, in Figure 200, router A spoofs the connection between C1 and S2 on port 3; router A spoofs the connection

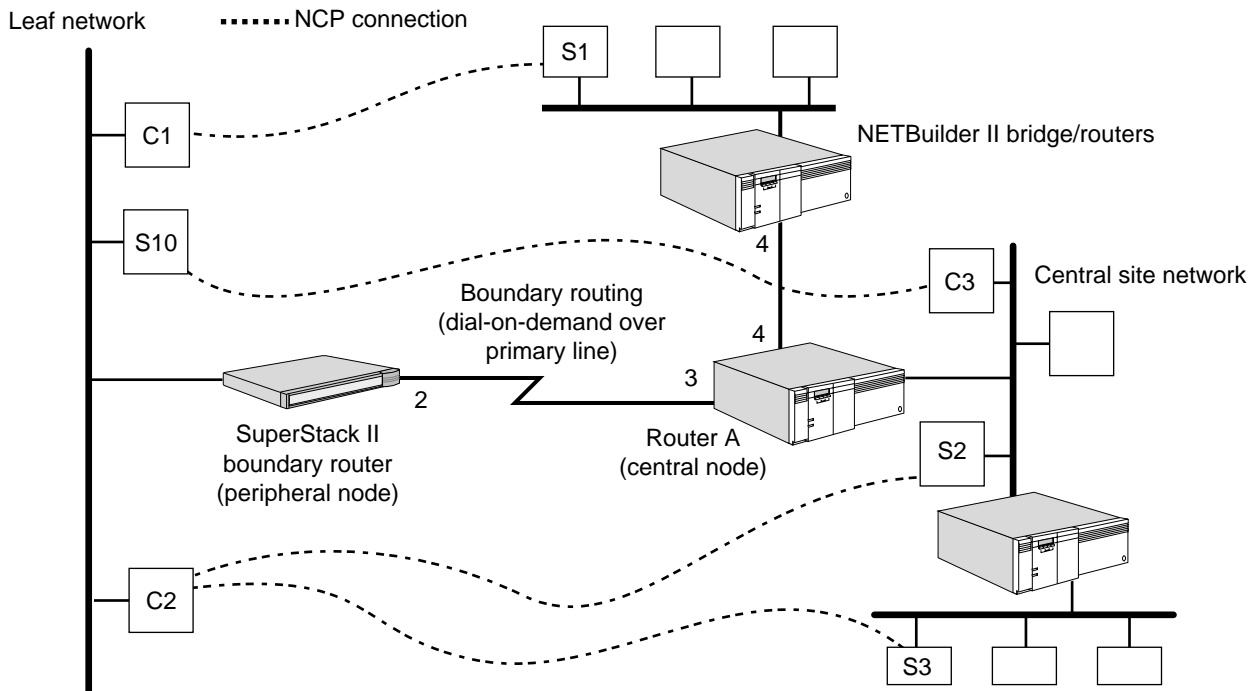
between C2 and S3 on port 3; router C spoofs the connections between C3 and S1. There is no spoofing between S3 and C5.

Figure 200 NCP Spoofing over DOD Lines in a Router-to-Router Configuration



In a Boundary Routing configuration, NCP spoofing can be used to support remote clients' access to central site servers, and also central site clients' access to remote servers. For example, in Figure 201, router A acting as a central node spoofs the connection between C1 and S1 on port 3; router A also spoofs the connections among C2, S2, and S3 on port 3. For the connection between S10 on the peripheral network and C3 on the central site, the SuperStack II boundary router peripheral node spoofs NCP keepalive packets on port 2.

Figure 201 NCP Spoofing over DOD Lines in a Boundary Routing Configuration



SPX1 Spoofing Lite over a DOD Link

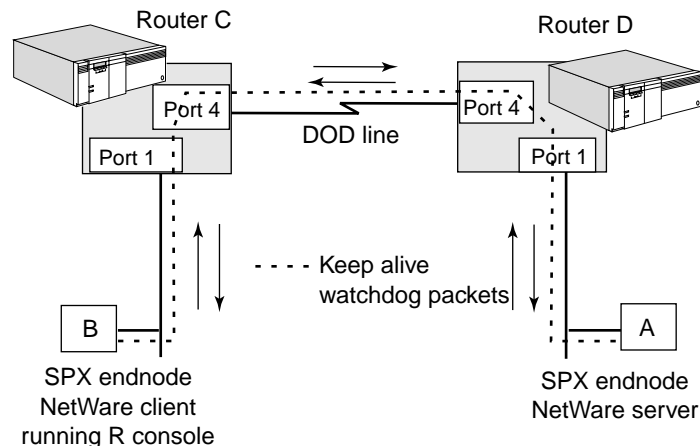
Sequenced Packet Exchange 1 (SPX1) is a transport-level connection protocol used by certain applications in the NetWare environment. An SPX1 client initiates a connection to an SPX1 host to transfer data with guaranteed delivery. This is done by transmitting an acknowledgment (ACK) packet. Depending upon the application, these data transfers can be bursty followed by long periods when the connection is quiet. To maintain this quiet period, SPX1 uses a process where an exchange of the ACK packets is performed by two SPX1 connection end nodes. These packets are referred to as *keepalive* or *watchdog* packets. When the watchdog packets are transmitted over a DOD line, the line is kept up unnecessarily, incurring extra costs. SPX1 Spoofing Lite is the 3Com solution to this problem. During the non-data transfer keepalive period, SPX1 Spoofing Lite spoofs SPX1 watchdog packets using an allocation window of 1. Only those applications such as Novell Rconsole or Lotus Notes that are not sensitive to this allocation setting are supported. It is also recommended that SPX1 Spoofing Lite be used only with ISDN DOD links. For some applications, the link must be brought up whenever there is SPX1 data to be exchanged within a short interval or the application will time out.

In Figure 202, SPX1 end nodes A and B exchange SPX1 watchdog packets to prevent their internal timers from expiring during times when there are no active SPX1 data transfers. If these timers expire, the connection will be aborted. When the connection is over a DOD link, the packets keep the line up.

When SPX1 Spoofing Lite is enabled with the `-IPX SPOOFCONTROL` parameter, all SPX1 packets to be forwarded out the DOD port are intercepted and processed, as follows:

- All data packets and all system packets except the watchdog packets are always forwarded across the DOD link.
- If the link is down, DOD brings the line up and the packets are forwarded.
- Watchdog packets are forwarded only when the link is up.
- When the link is down, as occurs during quiet times, watchdog packets will be discarded and spoofed.

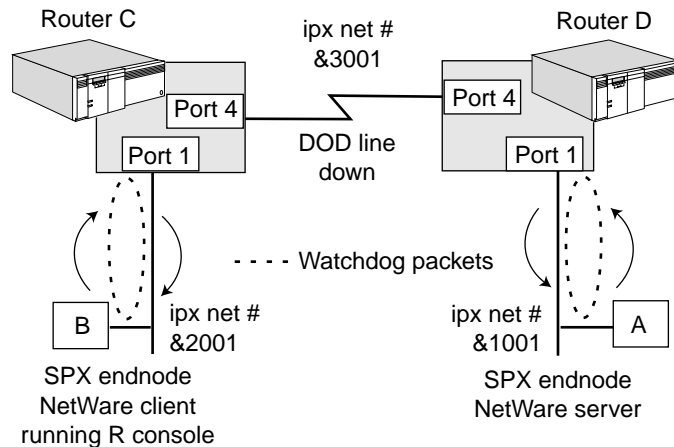
Figure 202 SPX Watchdog Packets over a DOD Link



In Figure 203, when the DOD link is down, watchdog packets from end node A are intercepted by router D and recycled as spoofed watchdog packets, then sent back out on port 1 to end node A. Watchdog packets from end node B are intercepted by router C and recycled as spoofed packets and sent back out on port 1 to end node B.

With SPX1 Spoofing Lite, watchdog packets are spoofed with a an SPX1 allocation window of 1. Only those applications that can handle this allocation size (Novell Rconsole, Lotus Notes, etc.) are supported. Use the `-IPX SPOOFCONTROL` parameter to enable or disable spoofing on each port. For non-DOD ports, spoofing does not apply and is always disabled.

Figure 203 Spoofing SPX Watchdog Packets



Supported Configurations

The SPX1 Spoofing Lite feature can be used with DOD in the following IPX network configurations:

- Router-to-router
- Boundary Router central node to Boundary Router peripheral node

Configuring SPX1 Spoofing Lite over a DOD Link See Figure 203 for an illustration of the following configuration. To configure SPX1 Spoofing Lite, follow these steps:

- 1 Enable IPX routing on router C and router D by entering:

```
SETDefault !1 -IPX CONTROL = ROute
SETDefault !4 -IPX CONTROL = ROute
```

- 2 Assign the IPX network numbers for port 1 and port 4 on router C by entering:

```
SETDefault !1 -IPX NETnumber = &2001
SETDefault !4 -IPX NETnumber = &3001
```

- 3 Assign network numbers for port 1 and port 4 on router D by entering:

```
SETDefault !1 -IPX NETnumber = &1001
SETDefault !4 -IPX NETnumber = &3001
```

- 4 Use incremental NRIP and SAP to reduce broadcast traffic on port 4 of both bridge/routers by entering:

```
SETDefault !4 -NRIP CONTROL = (Talk, Listen, NoPERiodic)
```

```
SETDefault !4 -SAP CONTROL = (Talk, Listen, NoPeriodic)
```

- 5 Enable SPX1 Spoofing Lite on both bridge/routers by entering:

```
SETDefault !4 -IPX SPOOFCONTROL = Spx1WatchDog
```

- 6 Verify that SPX1 Spoofing Lite is enabled by entering:

```
SHow !4 -IPX SPOOFCONTROL
```

Configuring SPX1 Spoofing Lite for the Boundary Routing Peripheral

Note SPX1 Spoofing Lite is disabled by default on the boundary routing peripheral node. To enable it, a special filter policy has to be configured. When configured, this policy enables SPX1 spoofing. When deleted, spoofing is again disabled. Filtering does not need to be enabled for this special policy to take effect. This special policy is to be used only on the peripheral node for enabling and disabling SPX1-spoofing. On the boundary routing central node, SPX1 spoofing is enabled or disabled using the -IPX SPOOFCONTROL parameter as described in the previous sections.

To enable SPX1 spoofing on the peripheral node and configure the special filter policy, follow these steps:

- 1 Add a user-defined mask called "SPX" for IPX packet type of 5 by entering:

```
ADD -fi ma spx ipx.pt = 5
```

- 2 Add the special SPX1 spoofing filter policy to enable spoofing by entering:

```
ADD -fi pol spoofspxl dod spx
```

- 3 To disable SPX1 spoofing on the peripheral node, enter:

```
DELEte -fi pol spoofspxl
```

Macros can be defined with these filter configurations to enable and disable spoofing. For example, a macro called SPOOFON could be defined which configures both the mask and the special policy. Another macro called SPOOFFOFF can be used to delete the policy.



On a DOD link with infrequent data traffic, the bridge routes may age out because of the infrequency of packets arriving from the central site to refresh those routes. In such a situation, the ageout timer should be disabled, or its value increased, for SPX1 spoofing to function properly.

- 4 To disable the ageout timer, enter:

```
SETDefault -brln cont = na
```

How the IPX Router Works

The 3Com IPX router provides network connectivity between Novell NetWare clients and servers located in the same building or in distant cities. The 3Com IPX router supports a subset of Novell's NetWare communication protocols that includes the IPX Protocol, NRIP, and SAP, NLSIP and minimal support of NetBIOS by propagating IPX WAN packets (packet type 0x14). However, the 3Com IPX router does not participate in any of NetWare Communication Protocol (NCP) or Sequenced Packet Exchange (SPX).

The 3Com IPX router can run over various types of data link media: Ethernet, token ring, FDDI, PPP, X.25, Frame Relay, and SMDS, and will support new media as they become available in the future. IPX has different types of encapsulation methods to run over various media. On Ethernet, four different encapsulation formats are available. The 3Com IPX router supports all of them, even

simultaneously (one physical network can be segmented into four different logical networks). Additional information on what encapsulation formats are available for each medium and how to configure them, and examples are in "Configuring Secondary Networks with Different Header Formats" earlier in this chapter.

IPX Router Features

The 3Com IPX router offers features including various NRIP and SAP policies, manageability via SNMP, static routing capability, and next-hop split horizon and NLSP. Various parameters are available to tune the IPX router to enhance network performance by reducing network overhead. For example, the nonperiodic (incremental) update mechanism reduces the number of NRIP and SAP updates on WAN interfaces and NLSP reduces routing updates throughout your IPX network. For conceptual information, see "Learning Routes and Service Information" later in this chapter. For procedural information, see "Controlling NRIP and SAP Updates" earlier in this chapter.

Each IPX host is uniquely identified with an IPX Internet address that consists of two parts:

- A four-byte IPX network number
- A six-byte IPX node address

The four-byte IPX network number (represented in hexadecimal) is assigned by a network administrator. The network number must be unique throughout the IPX Internet. Be careful not to assign duplicate networks; otherwise it causes network-wide confusion. When using NLSP, a portion of the network number also identifies the NLSP area.

The IPX node address (represented in hexadecimal) is permanently associated with each port and is not assignable except on the NetWare server's internal address. The 3Com IPX router has multiple ports and an internal network number. For instructions on assigning network numbers, see *New Installation for NETBuilder II Software*.

The static routing feature allows network managers to eliminate traffic associated with the route advertisements required for dynamic route learning, which frees bandwidth on slow serial data links for critical data traffic. IPX routing capability can still be achieved without sending a single NRIP update by setting the -NRIP CONTROL parameter to "NoTalk" and adding static routes on the port. Static routes can be especially cost-effective on any service where packet charges are enforced. One disadvantage of static routes is that these routes are not updated automatically. After being configured, they remain in the routing table until they are manually removed (even if the corresponding route no longer exists). For this reason, static routes are recommended only where the network topology remains constant. Another solution to this problem is to run NLSP.

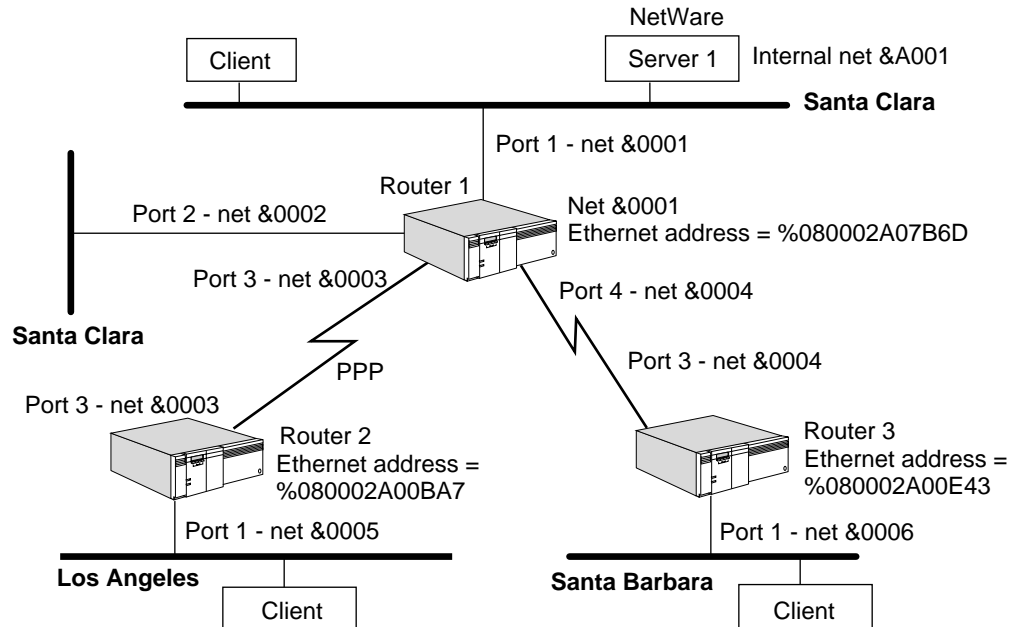
The following sections provide more detailed discussions of important concepts related to IPX routing.

Local and Wide Area Network Configuration

An IPX network must be configured on each local port on which IPX packets are to be received and sent. WAN ports using PPP may be configured with or without a network number, provided an internal network number has been configured. The port can be a local Ethernet, FDDI, token ring port or a serial line port on a wide area network, such as a point-to-point link or an X.25 link. Figure 204 is an

example showing a wide area router connecting two local Ethernet networks (Santa Clara) to two wide area networks (Los Angeles and Santa Barbara).

Figure 204 Wide Area Router Connecting Four IPX Networks



All IPX network numbers assigned must be unique within the IPX Internet.

Any physically attached network, Ethernet, or serial line is considered a directly connected network. If more than one serial line (path) is assigned to one port, that port is considered a single directly connected IPX network.

A router must look up the destination network in its routing table to determine where to route a packet. If the destination is on an attached network, the router can send it directly to the network. But if the destination is not directly connected, the router must route the packet to another router that is closer to the destination. The route to a remote network can be statically configured or dynamically learned through NRIP and NLSP routing protocols. For details, see "Enabling and Disabling Dynamic Learning and NRIP Updates" and "Adding and Deleting Static Routes" earlier in this chapter.

When two routers are located on the same network (that is, each of them has at least one port to the network) they are called *neighbors*.

You can set up routing without the assignment of IP subnets. This feature is called unnumbered links. Unnumbered links are useful only between two routers; in other words, you cannot connect a router to a host using unnumbered links. For more information about unnumbered links, see the Configuring IP Routing chapter.

Routing Tables To display the routing table, enter:

```
SHow -IPX AllRoutes
```

The AllRoutes parameter has three display options. The Short option (the default) displays only network numbers and hop counts. The Long option additionally

displays port numbers, network numbers, gateway addresses, hops, and costs. If you specify a network number for the NETnumber option, the port number, gateway address, hop count, and costs for the specified network are displayed. See the IPX Service Parameters chapter in *Reference for Enterprise OS Software* for information on the AllRoutes command.

Depending on the AllRoutes option selected, the routing table can include the following data, which determines how a packet is routed:

- Port number
This is the port associated with the attached network.
- NETnumber or "Default" label (which indicates a default route)
- Gateway address
This is the IPX address of the gateway to which a router must send the packet before the packet can be routed to the destination. For more information on the gateway address, see "Adding and Deleting Static Routes" earlier in this chapter.
- Number of hops between router and destination
The number of hops is equal to the number of gateways traversed.
- Costs associated with the route
- Status
- TTL
- Source

For each destination address, the router can support up to four routes (that is, four gateways). These routes, either learned or configured, are stored in the routing table. For information on how the router makes the routing decision, see "Routing Selection".

Service information is maintained in a server table. To display the contents of the routing table, enter:

SHow -IPX AllRoutes

The following display appears:

```
----- IPX Routing Table -----
00000001  5  0000000C  7  00000032  3  00000045  7  00022222  6
0002ED49  9  00034562  9  0004065B  7  00044C34  9  000464FB  9
00049001 10  00049003 10  0004AD0D4  9  0004CFEC  8  0000502E6 13
000502E8 12  00053376 11  0005419   8  00054669  5  00055A1E  8
```

SHow -IPX AllRoutes Long

The following display appears:

```
----- IPX Routing Table -----
NETnumber Gateway          Hops  Cost  Status  TTL  Source
00000001          5      6    UP     240   RIP
&DDDDD200%080002A078DB
0000000C          7     31    UP     240   RIP
&DDDDD200%080002A078DB
00000032          3      4    UP     240   RIP
&DDDDD200%080002A078DB
00000045          7      9    UP     240   RIP
&DDDDD200%080002A078DB
```

```
00022222                6      7      UP      240      RIP
&DDDDDD200%080002A078
DB
```

If you have a large routing table, you can specify a network number to verify its reachability using:

```
SHow -IPX AllRoutes <NETnumber>
```

Default Routes

When a router needs to route a packet destined for an address for which there are no entries in the routing table, it uses the default route if one exists. The network number &FFFFFFFE is reserved and represents the default route.

Default routes are important in building large, enterprise-wide networks. They allow an organization to perform route filtering at a border router and substitute the default routes with a single default route advertisement. A default route is useful over dial-on-demand lines, and can also be used as a backup route when the primary path is not available.

Effect on NRIP

NRIP recognizes and accepts the default route in NRIP advertisements received from other routers, enters it in the routing table, and propagates it if necessary. When forwarding IPX packets, an NRIP router forwards all unknown destination packets toward the default route.



The default route implementation for NRIP is not supported in software versions prior to version 8.2. All routers must be upgraded to software version 8.2 or later so that all NRIP routers recognize &FFFFFFFE as the default route and forward packets for unknown destinations toward it.

Effect on NLSP

An NLSP router can learn a default route in two ways:

- If there is an attached Level 2 NLSP router present, this router is considered the default route.
- Learning of network &FFFFFFFE from NRIP advertisements. This network number is imported and advertised to all other NLSP routers.

Forwarding unknown destination packets to the Level 2 router has higher precedence than forwarding an imported NRIP route. If there are no attached Level 2 routers, an NLSP Level 1 router forwards unknown destination packets toward the NRIP default route. If neither an attached Level 2 router nor an imported NRIP route is available, the NLSP Level 1 router drops the unknown destination packet.

Effect on SAP

The configuration of a default route has no effect on SAP advertisements, which list the network addresses of the available services. If the address is unreachable according to either an NRIP or NLSP update, the advertisement is dropped. This behavior is unchanged by the implementation of the default route.

Routing Selection

The IPX router selects the most efficient path for information. The most efficient path is the path that takes the least time to reach a destination. The amount of time needed to reach a destination is not configurable; it is based on the type of

interface your router uses. The faster the line your router uses, the less time it will take for a packet to reach its destination. For example, an Ethernet (10 Mbps) is faster than a T1 (1.54 Mbps) serial line; it takes less time for a packet to reach its destination via an Ethernet than a T1 serial line.

You can affect the amount of time it takes a packet to traverse a serial line by using a faster line and changing the baud rate using the `-PATH BAud` parameter. This method of affecting the time a packet takes to traverse a serial line is effective only if the clock source for the serial line uses the internal on-board clock oscillator (TestMode value of the `-PATH CLock` parameter). When two paths require the same amount of time for a packet to traverse (same cost delay), the router will select the path with the lowest hop count. The router selects the path learned first if they have the same hop count.

Learning Routes and Service Information

To report route changes to its neighbors and learn about other services that are available on the network, the router or server (file server, printer, etc.) sends NRIP and SAP updates, respectively. In a large IPX environment, these update packets create the major network overhead. The frequency of the updates depends on the settings of the UpdateTime and CONTROL parameters as follows:

| | |
|-----------------------------------|--|
| Periodic updates | By default, the router sends both NRIP and SAP updates at initialization and every 60 seconds (the default value of the UpdateTime parameter). When topology changes occur, updates are sent because Trigger is enabled by default. For details, see "Controlling NRIP and SAP Updates" and "Controlling Route and Service Aging" earlier in this chapter. |
| Nonperiodic (incremental) updates | The router sends NRIP and SAP updates only when topology changes occur. Incremental NRIP and SAP updates are enabled by the NoPEriodic values of the <code>-NRIP</code> and <code>-SAP CONTROL</code> parameters. The <code>-SAP CONTROL NoPEriodic</code> value is the default setting on WAN interfaces. Nonperiodic is the preferred method on a WAN because it uses less bandwidth. For details, see "Controlling NRIP and SAP Updates" earlier in this chapter. |

On LAN interfaces, the IPX router generates regular NRIP and SAP updates every 60 seconds. On slow WAN links, these NRIP and SAP updates can take up the bulk of the network traffic. In order to minimize network overhead, the router pays special attention to NRIP and SAP updates on WAN interfaces. By using the nonperiodic (incremental) update mechanism (enabled if the `-SAP CONTROL` parameter is set to `NoPEriodic`), the router does not send any NRIP or SAP updates over WAN interfaces except those containing new information after the system is initialized. `NoPEriodic` updates can substantially reduce network overhead over WAN links and can also be used on LAN interfaces if the NetWare servers on that network also support nonperiodic updates. All routers and servers on the same network should use the same update mechanism (periodic or nonperiodic).

You can also control if and how the router advertises routes to a neighbor from which it learned the same route. For details, see "Controlling NRIP and SAP

Updates” earlier in this chapter. Another solution to routing overload is to use NLSP.

Regular route update packets contain the following types of information:

- The networks it can reach
- The number of hops and the amount of time associated with each network it can reach

For information on routing table entries, see “Routing Tables” earlier in this chapter.

Regular SAP updates packets contain the following types of information:

- Server type
- Name of the server
- Network address of the server
- Number of hops associated with each server

For information on service table entries, see the next section.

Server Tables

Server information is maintained in a server table. To display the contents of the server table, enter:

```
SHow -IPX AllServers
```

or

```
SHow -IPX AllServers Long
```

Adding “Long” to the command displays gateway information along with the server table contents.

If you have a large server table, you can specify a server name to display single server information using the SHow -IPX AllServers “<string>” syntax. An entry in the server table times out in the same way as a routing table entry (see “Controlling Route and Service Aging” earlier in this chapter for details). When a server becomes unreachable, an update packet with this information is sent out immediately (see “Controlling NRIP and SAP Advertisements” earlier in this chapter for details).

Network Reachability

When dynamic learning of routes is enabled, a router learns new routes from RIP update packets broadcast by its neighbors. The following are considered *reachable* when a router broadcasts its RIP update packets:

- Directly connected networks
- Static routes
- Dynamic routes learned through RIP that are currently in the routing table (that is, dynamic routes that have not timed out)

Solving the Slow Convergence Problem with Split Horizon

All routers need to learn of new routes and discard obsolete routes immediately. That is, the contents of their respective routing tables converge rapidly so that all routing tables always contain correct information. An undesirable side effect of NRIP is the possibility that the time during which the unreachable network is

thought to be reachable is prolonged. One solution to this problem of slow convergence is called *split horizon*.



The following explanation describes split horizon for NRIP, but also applies to SAP.

The 3Com IPX router offers two methods for achieving split horizon: split horizon per network number and split horizon per neighbor, also known as next-hop split horizon. In a WAN environment, next-hop split horizon eliminates the need for a fully meshed network. With next-hop split horizon, the router learning of new routes records the IPX Internet address (network number and host address) of the advertising router and applies the split horizon algorithm per neighbor. Connectivity between different remote offices in a nonmeshed WAN topology can be maintained with next-hop split horizon while split horizon per network always expects a fully meshed topology.

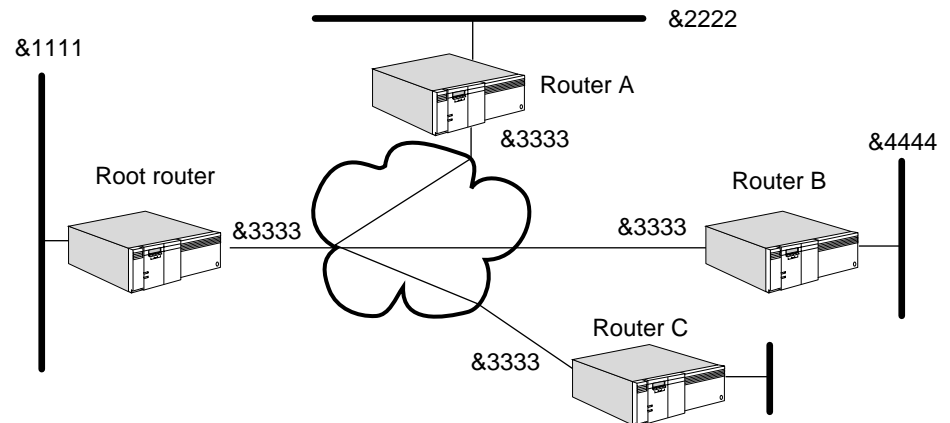
Figure 205 shows a nonmeshed network on which router R is the root router and routers A, B, and C are remote routers that are configured as neighbors on router R. (This example applies to Frame Relay, ATM, and X.25 networks.) When both advertise and receive neighbor policies are disabled, split horizon per network takes effect. In this case, Router R excludes from its RIP updates on network &3333 all routes (&2222, &3333, &4444, and &5555) learned from network &3333 if you select the NoPOison option of the -NRIP CONTrol parameter. If you select the POison option, router R includes routes but sets their hop count to 0xFFFF.

By applying next-hop split horizon, see “How the IPX Router Works” earlier in this chapter for information about next-hop split horizon, router R does not advertise network &2222 to router A, because it learned of &2222 from router A (identified by router A's IPX address) or include it, but set its hop count to 0xFFFF depending on the POison/NoPOison option. For the same reason, router R does not advertise network &4444 to router B, nor does it advertise &5555 to router C, because it learned of those networks from those routers.

On Frame Relay, ATM, or X.25 networks, you must configure the host-to-media address mappings (ADDRESS parameter). On Frame Relay networks, the bridge/router performs automatic DLCI learning and automatic host-to-DLCI address learning based on incoming IPX packets. Manually configure the host-to-DLCI address mapping because incoming IPX packets are not always guaranteed.

The host-to-media mappings (either configured or automatically learned) are used for transmitting NRIP and SAP advertisements. For NLSP, the host-to-media mappings are used for establishing adjacencies. The mapping information is useful regardless if the topology is full- or partially meshed.

LAN Networks On a LAN, you do not need to configure neighbors, but if neighbor policies are enabled and neighbors are configured, NRIP unicasts the updates to each neighbor. If neighbor policies are disabled, NRIP broadcasts the updates over the LAN.

Figure 205 Route Advertisement Over Nonmeshed Frame Relay Network

No additional configuration is necessary to use the next-hop split horizon feature. It is automatically configured when neighbors are configured.

Solving the Slow Convergence Problem with Poison Reverse

Poison reverse or no poison reverse is configurable via the POison and NoPOison values for the -NRIP CONTROL parameter.

If poison reverse is enabled, the router advertises all routes to all neighbors, but when advertising a route to a neighbor that has advertised the same route, the router sets the hop count to infinity (0xFFFF) to prevent the recipient from adding the route to its routing table. Poison reverse speeds convergence but adds to network overhead.

If poison reverse is disabled, the router omits routes learned from one neighbor from NRIP updates sent to that neighbor. No poison reverse has the advantage of minimizing network overhead in large network configurations at the expense of slower convergence.

Route, Service, and Neighbor Policies

Route policies can be used to limit the view of the IPX Internet as seen from a specific segment, suppress reachability to selected networks in the Internet from specific segments, and provide security or segment isolation. Route policies also allow control of the propagation of routes to areas of the Internet where these routes are not needed, with the effect of controlling the sizes of the routing tables.

Route policy applies to the following events:

- NRIP updates received from other routers, called receive policy for routes.
- NRIP updates sent by the router, called advertise policy for routes. The NRIP updates are broadcast at regular intervals or whenever there are changes to the routing table.
- NRIP responses sent by the router whenever a NRIP request is received from a specific IPX host. The advertise policy can also be used to answer NRIP requests from a specific IPX host.

Service policies can be used to limit access to service from specific segments in the Internet, provide security or access-control, and reduce overhead by not advertising unnecessary resources. For example, access to a print server can be restricted to the segment where that printer's designated users are located, and the print service on that server is not advertised to the rest of the IPX Internet. Similar to route policies, the size of the service-related tables can be controlled by advertising only those services that need to be made available.

Service policy applies to the following events:

- SAP updates received from other routers, called receive policy for services.
- SAP updates sent by the router, called advertise policy for services. SAP updates are broadcast at regular intervals or whenever there are changes to the SAP table.
- SAP responses sent by the router whenever a SAP request is received from a specified IPX host. The advertise policy can also be used to answer SAP requests from a specified IPX host.

Neighbor policies are used to ensure that the router accepts routing information from and sends routing information to routers that are designated as neighbors.

Neighbor policy applies to the following conditions:

- The source or originator of NRIP and SAP updates. The neighbor is identified by the MAC address of the originator. The neighbor identification restricts information received.
- The destination of IPX hosts identified by the IPX network number and its MAC address. The neighbor identification selectively sends NRIP and SAP updates when responding to NRIP requests or SAP queries. If dynamic neighbors are enabled, the NRIP and SAP updates and responses are sent to all known neighbors.

Neighbor policies affect NRIP and SAP updates received from neighboring routers, regular and triggered NRIP and SAP updates sent to neighboring routers, and NRIP and SAP responses sent because of specific queries made by a client. If NRIP and SAP responses are sent because of a query by a client and the requesting client is not in the neighbor list that the router uses for sending NRIP and SAP updates, then no response is issued.

Policy Control

You can control route, service and neighbor policies as follows:

- You can disable policies during network operations.
When a policy is disabled, the configured items corresponding to that policy are retained but are not used. Disabling policies at runtime is done through the PolicyControl parameter.

- You can configure the router to override the policies when responding to specific route and service requests using the PolicyControl parameter.

That is, the policies are used for regular updates and triggered updates that are sent by the router during normal operation, but regular updates and triggered updates are overridden when the router responds to NRIP and SAP requests. The response to NRIP and SAP requests are sent directly to the requestor.

- You can configure the router to derive the routes being advertised on any specific interface from the configured service policies for that interface.

Route advertisement decisions can be made using the service policy list. When service advertisement policies are configured and enabled, while route advertisement policy is enabled, but no route policies are explicitly configured, then the router policies are derived from the service policies. That is, if a service is identified on a network for inclusion in the SAP advertisement, then the network is also included in the NRIP advertisement.

- You can configure policy lists (lists of routes that are filtered out of NRIP updates received on a specified interface) as inclusion or normal policies, or exclusion or inverse policies.

Inclusion policies specify those items in the lists for inclusion in the NRIP updates and all other list items are excluded or filtered. Exclusion policy specifies the items for exclusion or filtering and all other items in the list are included in the NRIP updates. Policy lists can be applied to all parameters except AdvToNeighbor by prefixing the policy items with the tilde (-) character which indicates excluded list items.

Route Receive Policy

You can use the route receive policy to restrict the routes accepted from NRIP updates received on a specified port before the update is processed.

To restrict the routes that are accepted from NRIP broadcasts follow these guidelines:

- Use the ReceivePolicy parameter to identify the networks or routes that you want to include or exclude from the router's routing table when they are received in a NRIP update on the interface specified.

Routes are identified by network number. Network number ranges can be specified to include or restrict a group of networks in the ReceivePolicy parameter.

- Use the ReceivePolicy attribute of the PolicyControl parameter to enable route receive policy.

If the PolicyControl attribute ReceivePolicy is set with no route receive policies configured, the router will not accept any routes that are being advertised to it by other routers on the specified interface.

Route Advertisement Policy

To restrict the routes that are advertised on a specified interface through regular and triggered updates, and those that are sent in NRIP responses to specific NRIP requests, follow these guidelines:

- Use the AdvertisePolicy parameter to identify the networks or routers that must be included in or excluded from NRIP updates or NRIP response broadcast from the specified interface.

Routes are identified by network number. Network number ranges can be specified to include or restrict groups of networks in the AdvertisePolicy parameter.

- Use the AdvPolicy attribute of the PolicyControl parameter to enable route advertise policy.

To restrict the routes advertised on a specified interface through regular and triggered updates, without causing restriction of any routes that are otherwise included in NRIP responses to specific NRIP requests, follow these guidelines:

- Use the `AdvertisePolicy` parameter to identify the networks or routes that must be included in or excluded from regular and triggered NRIP updates that are sent out the specified interface.

Routes are identified by network number. Network number ranges can be specified to include or restrict groups of networks in the `AdvertisePolicy` parameter.

- Use the `PolicyControl` parameter to enable route advertise filtering by setting the attribute `AdvPolicy`.
- Use the `PolicyControl` parameter to enable the policy override option for NRIP responses by setting the `PolicyOverride` attribute.

The `PolicyOverride` option applies to both NRIP responses and to service queries. To determine the routes that you want to include in regular and triggered updates and responses to specific NRIP requests from the service policies that are configured for a specified interface, follow these guidelines:

- Use the `AdvertisePolicy` parameter to identify the services that are required for inclusion or exclusion from SAP updates and responses.

Routes are identified by network number. Network number ranges can be specified to include or restrict groups of networks in the `AdvertisePolicy` parameter.

- Use the `PolicyControl` parameter to activate the service policies by setting the `AdvPolicy` attribute.
- Use the `PolicyControl` parameter to enable the route advertisement policy by setting the `AdvPolicy` attribute.

This method of determining the route policies from the service policies works only when the service advertisement policy is enabled. If the `PolicyControl` attribute `AdvPolicy` is set, no route advertise policies are configured, and there are no effective service advertise policies, then the router will not advertise any routes that are in its routing table to other routers on the specified interface.

Service Receive Policy

To restrict services from being accepted from SAP updates received on a specific port before the update is processed, follow these guidelines:

- Use the `ReceivePolicy` parameter to identify the services that are received in a SAP update on the specified interface that you want included in or excluded from the router's routing table.

A service is identified by the network where the service is located, the host's MAC address, or the name of the server where the service and service type are located. Network number ranges and wildcards for network numbers, server host address or name and service types can be used to group services in the `ReceivePolicy` parameter.

- Use the `PolicyControl` parameter to enable service receive policy by setting the `RcvPolicy` attribute. If the `PolicyControl` attribute `RcvPolicy` is set, and there are no service receive policies configured, then the router will not accept any services that are being advertised to it by other routers on the specified interface.

Service Advertisement Policy

To restrict the services that are advertised from a specified interface through regular and triggered updates and those that are sent in SAP responses to specific SAP requests, follow these guidelines:

- Use the `AdvertisePolicy` parameter to identify the services you want included in or excluded from SAP updates or SAP responses sent out of the specified interface.

A service is identified by the network where the service is located, the host's MAC address, or the name of the server where the service and service type are located. Network number ranges and wildcards for network numbers, server host address or name and service types can be used to group services in the `ReceivePolicy` parameter.

- Use the `PolicyControl` parameter to enable the service advertise policy by setting the `AdvPolicy` attribute.

To restrict the services that are advertised from a specified interface through regular and triggered updates, but not restricting any services that are included in SAP responses to specific SAP requests, follow these guidelines:

- Use the `AdvertisePolicy` parameter to identify the services that must be included in or excluded from the regular and triggered SAP updates broadcast from the specified interface.

A service is identified by the network where the service is located, the host's MAC address, or the name of the server where the service and service type are located. Network number ranges and wildcards for network numbers, server host address or name and service types can be used to group services in the `ReceivePolicy` parameter.

- Use the `PolicyControl` parameter to enable service advertise filtering by setting the `AdvPolicy` attribute.
- Use the `PolicyControl` parameter to enable the policy override option for SAP responses by setting the `PolicyOverride` attribute.

If the `PolicyControl` attribute `AdvPolicy` is set, and there are no service advertise policies configured, then the router will not advertise any services that are in its SAP table to other routers on the specified interface.

Neighbor Policy

To restrict the number and identity of routers that the listening router should accept NRIP and SAP updates from, follow these guidelines:

- Use the `RcvFromNeighbor` parameter to identify the routers.
Neighbors are identified by their host's MAC address in the `RcvFromNeighbor` parameter.
- Use the `PolicyControl` parameter to enable the neighbor policy for received NRIP and SAP updates by setting the `RcvFromNbr` attribute.

If the `PolicyControl` attribute `RcvFromNbr` is set, and a list of neighbors to receive from has not been configured, then none of the NRIP and SAP updates received are accepted.

To restrict the number and identify the neighbors the sending router can broadcast NRIP and SAP updates to, and those the router can accept NRIP requests or SAP queries from, follow these guidelines:

- Use the `AdvToNeighbor` parameter to identify the neighbors.

The router can be configured to send a unicast copy of the NRIP and SAP update. Each neighbor is identified by the IPX network number and its MAC address in the AdvToNeighbor parameter.

- Use the PolicyControl parameter to enable the neighbor policy for advertisement of NRIP and SAP updates and for responses to NRIP requests and SAP queries by setting the AdvToNbr attribute.

If the PolicyControl attribute AdvToNbr is set, and dynamic neighbors are enabled, all NRIP and SAP updates are sent to all known neighbors individually.

If the PolicyControl attribute AdvToNbr is set, and no neighbors are identified, then NRIP and SAP updates will not be broadcast from the specified interface and there will be no response to any requests or queries received on that interface.

Novell Service Types

When setting IPX NLSP, NRIP, and SAP policies, you may need information for Novell Service Types available on file servers. Table 50 lists the Novell Service Types and the object type (in hex) that should be used.

Table 50 Novell Service Descriptions

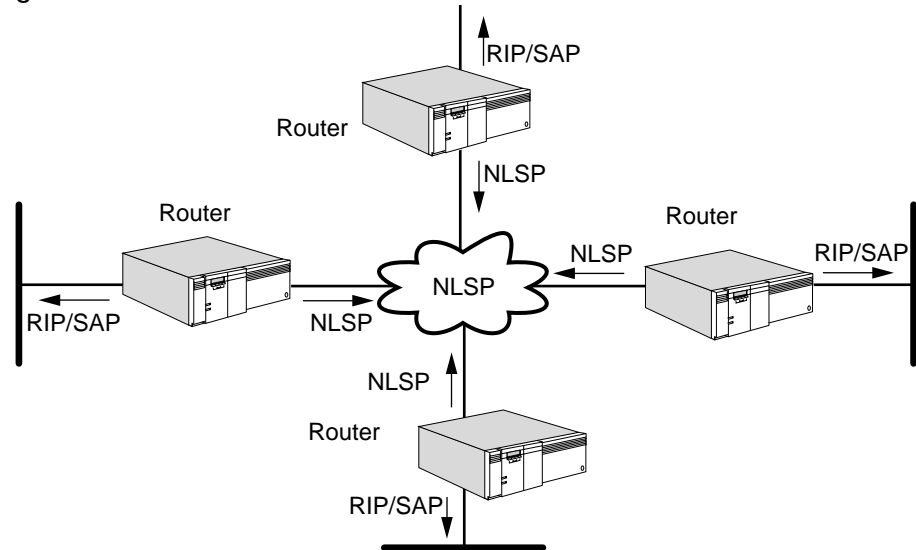
| Description | Object Type (hex) |
|-----------------------------|-------------------|
| User | 0x0001 |
| User Group | 0x0002 |
| Print Queue | 0x0003 |
| File Server | 0x0004 |
| Job Server | 0x0005 |
| Gateway | 0x0006 |
| Print Server | 0x0007 |
| Archive Queue | 0x0008 |
| Archive Server | 0x0009 |
| Job Queue | 0x000A |
| Administration | 0x000B |
| NAS SNA Gateway | 0x0021 |
| NACS | 0x0023 |
| Remote Bridge Server | 0x0024 |
| Bridge Server | 0x0026 |
| TCP/IP Gateway | 0x0027 |
| Gateway | 0x0029 |
| Time Synchronization Server | 0x002D |
| Archive Server SAP | 0x002E |
| Advertising Print Server | 0x0047 |
| Btrieve VAP 5.0 | 0x004B |
| SQL VAP | 0x004C |
| XTREE Network Version | 0x004D |
| Btrieve VAP 4.11 | 0x0050 |
| Print Queue User | 0x0053 |
| WANcopy Utility | 0x0072 |

Table 50 Novell Service Descriptions (continued)

| Description | Object Type (hex) |
|--------------------------|-------------------|
| TES - NetWare for VMS | 0x007A |
| NetWare Access Server | 0x0098 |
| Portable NetWare | 0x0107 |
| NetWare 386 | 0x0107 |
| Communications Executive | 0x0130 |
| NSS Domain | 0x0133 |
| NetWare 386 Print Queue | 0x0137 |
| NetWare 386 SAA Server | 0x0304 |
| Wildcard | 0xFFFF |

NLSP Routing

The NLSP routing protocol was developed by Novell to provide network layer connectivity in IPX networks. NLSP provides faster convergence and less overhead than other routing protocols by using a link-state-based routing algorithm. NLSP is designed as a router-to-router protocol. Clients and servers are not expected to participate in the NLSP packet exchange and continue to expect RIP and SAP updates. NLSP, RIP, and SAP coexist on the same internetwork: NLSP manages route and server information exchanges between routers and RIP and SAP advertise route and server information to end systems. Figure 206 shows the NLSP coexistence with RIP and SAP.

Figure 206 NLSP and RIP/SAP Coexistence

Hierarchical Routing

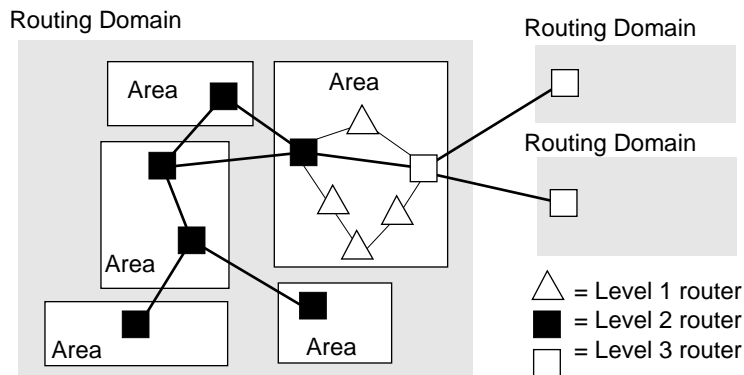
NLSP provides a hierarchical network topology that reduces overhead and allows the internetwork to scale because the NLSP routing overhead is confined to a particular area. Routing domains provide administrative boundaries in the internetwork.

In the NLSP hierarchical topology, networks are organized into areas, and areas are grouped into multiple routing domains as shown in Figure 207. A routing domain is a stand-alone administrative entity (such as a company, a university, or an agency). Routing domains are interconnected by Level 3 routers. Each routing domain can be further subdivided into multiple areas. An area can be a department, a building, or a group of highly connected and functionally related workstations or servers. An area can be as small as a single LAN, or as large as several hundred networks and hundreds of routers. Areas are interconnected by Level 2 routers. All routers within an area are Level 1 routers.



The current implementation for NLSP operates within an area only.

Figure 207 NLSP Hierarchical Routing



A single router is the minimum area that can be formed. The maximum area can contain hundreds of routers and networks, however, because memory overhead on a router is proportional to the size of its home area, the real size of an area will be conservative.

All routers belonging to the same area must be directly interconnected through physical paths. Any router must be able to reach any other router in the same area through intra-area routes by going through other routers belonging to the same area.

Routers in an NLSP environment form adjacencies with each other, and exchange information with adjacent routers about the status of their connected networks through link state packets (LSPs). The LSPs are used to build link-state databases, which are synchronized between adjacent routers to ensure accuracy. The LSPs are flooded throughout the area and all routers maintain identical detailed information about the topology of that area. If a network in that area changes status, an LSP are flooded quickly throughout the area to record the change.

Area Addressing

Each router must identify one to three area addresses, which are communicated to adjacent routers in the LSP packets and are also reflected in the network number portion of the IPX address. The IPX network number is a 32-bit integer, of which some bits identify the area and others identify the network within that area. The identification of both the value and length of the area address is configured in the -NLSP areaAddress parameter using:

```
ADD -NLSP AreaAddress <net> <mask>
```

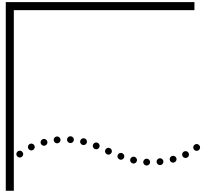
Each of the <net> and <mask> fields are 32-bit integers, the <net> field representing the value of the area address and the <mask> field representing how many of the 32 bits in the IPX network number are used to identify the area. For example:

```
ADD -NLSP AreaAddress 12345600 FFFFFFF00
```

A router can be configured with up to three area addresses, in which case a single area still exists but has three possible identifiers. A maximum of three area addresses are allowed in any area. If there exists more than three addresses within an area, the higher area addresses are dropped.

IPX Routing Terms

spoof A process that allows the bridge/router to respond to incoming NCP KeepAliveRequest or SPX1 watchdog packets that are to be routed over a DOD line, by sending a packet to the originating server of the request on behalf of the intended client. Spoofing occurs only when the DOD path is down to prevent the DOD path from constantly being brought up and down due to the transmission of packets from the server.



CONFIGURING APPLE TALK ROUTING

This chapter describes how to configure, customize, and troubleshoot a basic AppleTalk router.



For conceptual information, see "How the AppleTalk Router Works" later in this chapter.

Setting Up a Basic AppleTalk Router

This section describes how to set up a basic AppleTalk router. After you perform these minimum configuration steps to configure your AppleTalk router, you can use the default values of other parameters, or you can further customize the AppleTalk router as described in "Customizing the AppleTalk Router" later in this chapter.

Prerequisites

This section assumes that you have logged on to the system with Network Manager privilege and set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Before setting up an AppleTalk router, create a router plan. The router plan will help you determine how the AppleTalk internetwork will look and which router will "seed" each network. Remember that each internetwork is unique. There are no absolute rules that govern placement of seed routers in an internetwork.

Creating a Router Plan

To create a router plan, follow these steps:

- 1 Make a diagram of your proposed AppleTalk internetwork.
Include the physical network layout and connecting points (for example, routers and bridges) in your diagram. For an example of a diagram, see Figure 211.
- 2 For each network, determine the following information:
 - The number of AppleTalk devices (for example, workstations, servers, and printers) present and projected.
 - The quantity of network numbers sufficient to satisfy capacity requirements (up to $n \times 253$ devices can be supported, where n is the number of network numbers in the range). 3Com recommends leaving gaps between network number ranges in order to accommodate network growth.
 - The number of zones and names needed and which devices will be in each zone for those networks with more than one. You will also need to identify which zone will be the default zone of the network.

- 3 Create a table of your router seeding plan, indicating which router will seed each network.

For definitions of seed and nonseed routers, see “Related Information” later in this chapter.

When you complete this table, you should have a record of all network number ranges in use, all zones in use, and which AppleTalk routers define zones and network numbers for each connected network.

In the simplest router seeding plan, you may pick one bridge/router per physical network as the seed router for that network. A single bridge/router can seed multiple networks (up to the maximum number of ports available).

An alternative plan is to set up multiple seed routers that supply identical information for a network. If the seed router hardware stops functioning and all seed routers have to be rebooted, you will not have to configure a new router to replace the disabled router at an inconvenient time. Another router with redundant seeding information can fill the role of seed router immediately. For more information, see “Setting Up Multiple Seed Routers” later in this chapter.

- 4 For maintenance purposes, you should create a database from your router seeding plan. Include the following information:

- Router location

Router location includes physical location and router name. The router name can be common to all names of ports (as specified by the RouterName parameter) on the router.

- Router type and version

- Networks connected to the router with the following information for each:

- Cabling identification

- Port type (EtherTalk, TokenTalk, LocalTalk, Fiber Distributed Data Interface (FDDI), Point-to-Point Protocol (PPP), X.25, Switched Multimegabit Data Service (SMDS), or Frame Relay)

- Seed information, if configured: network range, zone list, and default zone

- Data link address for each router port (media access control (MAC) address, X.25 Data Terminal Equipment (DTE), Frame Relay, Data Link Connection Identifier (DLCI), SMDS individual and group address)

Procedures This section provides information on configuring local and wide area networks.

Configuring for Local Area Networks

This section provides information on how to configure AppleTalk routers on Ethernet, token ring, and FDDI networks.

Your router plan will help you determine which routers need to be configured as seed routers. All other routers not configured as seed routers must be configured as nonseed routers. This section provides procedures on how to set up your router as a seed or nonseed router.

To set up a seed router, follow these steps:

- 1 Specify the range of network numbers that can be used on the cable to which the router port is attached using:

```
SETDefault !<port> -AppleTalk NetRange = <network-range>
```

- 2 If most end nodes on a cable will be in a single zone, use that zone as the default. Specify the default zone name for the network attached to a port using:

```
SETDefault !<port> -AppleTalk DefaultZone = "<zone-string>"
(1-32 char)
```

- 3 Specify additional zone names for nodes to be placed in different zones using:

```
ADD !<port> -AppleTalk ZONE "<zone-string>" (1-32 char)
```

All seed routers must have the same net range, zone list, and default zone.

- 4 Enable AppleTalk routing on the port using:

```
SETDefault !<port> -AppleTalk CONTROL = (RRoute, AppleTalk, SeedingAllowed)
```



This step must be performed after network number range and zone information are configured.

To set up a nonseed router, enable AppleTalk routing and disable seed router capability on a particular port using:

```
SETDefault !<port> -AppleTalk CONTROL = (RRoute, AppleTalk,
NoSeedingAllowed)
```

Repeat this step for other ports if appropriate.

For complete information on all parameters used in these procedures, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

Configuring for Wide Area Networks

Routing AppleTalk over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies.

If you plan to route AppleTalk over Frame Relay, ATM DXI, or X.25 in a partially meshed or nonmeshed topology, you must make certain that static AppleTalk address mappings are defined. Defining these mappings enables the next-hop split horizon feature. For complete information on configuring AppleTalk routing over Frame Relay, ATM DXI, or X.25, including a discussion on fully meshed, partially meshed, and nonmeshed topologies and next-hop split horizon, see the Configuring Wide Area Networking Using Frame Relay chapter, the Configuring Wide Area Networking Using the ATM DXI chapter, and the Configuring Wide Area Networking Using X.25 chapter.

Routing AppleTalk over SMDS is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your AppleTalk router to

perform routing over SMDS, see the Configuring Wide Area Networking Using SMDS chapter.

PPP links should be configured as non-AppleTalk data links. No static configuration is required. For more information, see "Setting Up AppleTalk Routing over a Non-AppleTalk Data Link" later in this chapter.

For information on wide area networking using Integrated Services Digital Network (ISDN), see the Configuring Wide Area Networking Using ISDN chapter.

Related Information

AppleTalk routing involves the following two types of routers:

- Seed routers

These routers serve as initial information and query points for other routers and end systems on AppleTalk networks. Each network cable, or set of bridged segments that are to be treated as a single AppleTalk network, must have at least one seed router. Seed routers require more configuration than nonseed routers and should be the first AppleTalk devices booted on a network. It is suggested that multiple routers be configured with identical seed information for redundancy.

- Nonseed routers

These routers require a minimum of configuration steps. Nonseed routers connected to AppleTalk networks must obtain information such as network numbers and zone lists from another router acting as a seed router on a connected network. The specific router that provides information to a new nonseed router is usually the first discovered by the new router.

3Com routers can also be used to route AppleTalk across non-AppleTalk backbone networks or point-to-point wide area links. These routers do not need to share seed information; they only share routing and zone information about the AppleTalk networks of which they are aware. See "Setting Up AppleTalk Routing over a Non-AppleTalk Data Link" later in this chapter.

After enabling routing on a port or when booting the bridge/router, a `SHoW` command executed before the AppleTalk router has completed the initialization phase may display parameter values that imply that the router is still configured to `NoRoute`. The `SHoW -AppleTalk DIAGnostics` command gives you the current state of each port.

A router can be a seed router on all ports; however, a router does not have to be a seed router for all the ports over which AppleTalk is routed. For example, a router with connections to networks over three ports may serve as a seed router for two of these and not as a seed router for the third.

During configuration, you must decide whether or not a port will be seeding. If it is, you must configure seeding information. If it is not to be a seed router, it is assumed that the connected network will be seeded by another AppleTalk router attached to the same network.

A seed router port must be configured to contain the following information:

- Network number range (the `NetRange` parameter)
- A list of one or more AppleTalk zones (the `ZONe` parameter)

- The default zone for the network if more than one zone is configured (the DefaultZone parameter)

The CONTROL parameter options also control how seed information is used and provide inter-router seed information validation. For more information, see “Port Startup Operations” later in this chapter.

Verifying the Configuration

To verify that the routers you configured are recognized by the network and are functional, follow these steps:

- 1 Check for possible problems using:

```
SHoW !<port> -AppleTalk DIAGnostics
```

The router displays a variety of information, depending on conditions detected by the software. For a general description of information available through the DIAGnostics parameter display, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

- 2 Check the routing table by entering:

```
SHoW -AppleTalk AllRoutes
```

The routing table displays all the networks to which a router has access directly or indirectly. Make sure that all expected networks are listed. Check that the expected *next routers* to the networks listed appear in the routing table. You may need to see your planning documents to associate data link addresses with routers.

- 3 Display the mapping information between zone names and network numbers and between network numbers and zone names by entering:

```
SHoW -AppleTalk ZoneNetMapping
```

```
SHoW -AppleTalk NetZoneMapping
```

For the mapping information between zone name and network number, the router displays a list of all zones and their associated networks on the AppleTalk internetwork that are known to the router. Make sure all expected zones are present. It usually takes a minute or less to acquire network and zone information, but may take longer depending on the size of the AppleTalk internetwork.

For the mapping information between network number and zone name, the router displays a list of associated zones for each known network. Make sure that all zone lists are complete (check the display for messages.)

Check these displays for accuracy. If a discrepancy appears, you must check and adjust the zone lists for seed routers directly connected to the networks in question. See “Changing a Zone List” later in this chapter.

- 4 Check the AppleTalk-specific configuration using:

```
SHoW !<port> -AppleTalk CONFIguration
```

```
SHoWDefault !<port> -AppleTalk CONFIguration
```

The SHoW configuration command displays live values. The SHoWDefault command displays the values you have configured.

To obtain seed router status for an interface, the network range and at least one zone need to be specified for the network zone list. If there are unexpected results, enter:

```
SHoW -AppleTalk DIAGnostics
```



In addition to performing checking procedures, the AppleTalk router is also an AppleTalk echo protocol responder. Reachability can be checked from another AppleTalk router on the AppleTalk internetwork using the APING command. For more information, see the *Commands* chapter in *Reference for Enterprise OS Software*.

Getting Statistics

To gather statistics, enter:

```
SHoW -SYS STATistics -AppleTalk
```

For a sample display and an explanation of the display, see the Statistics Displays appendix.

You can collect statistics for a specific time period by using the -SYS SampleTime and -SYS STATistics parameters. For more information, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*.

Troubleshooting the Configuration

If you are unable to make connections to nodes within a local area or nodes in other areas after setting up the router, review the following troubleshooting procedure. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier for assistance.

To troubleshoot your configuration, follow these steps:

- 1 Display diagnostic information stored by the router by using:

```
SHoW !<port> -AppleTalk DIAGnostics
```

The router displays a variety of information, depending on conditions detected by the software. For a general description of what is available through the DIAGnostics parameter display, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

- 2 If the diagnostic information displayed indicates that a port is down, enter:

```
SHoW -PORT CONFIguration
```

```
SHoW -PATH CONFIguration
```

- 3 Check the AppleTalk-specific configuration using:

```
SHoW !<port> -AppleTalk CONFIguration
```

```
SHoWDefault !<port> -AppleTalk CONFIguration
```

The SHoW configuration command displays live values. The SHoWDefault command displays the values you have configured.

Check that the displayed configuration is the correct one for this router.

- 4 Check for a misconfigured port owner using:

```
SHoW [!<port>] -PORT OWNEr
```

- 5 Check whether the network you are trying to reach is in the AppleTalk routing table using:

```
SHoW !<port> -AppleTalk AllRoutes <network range>
```

If the network you are trying to reach is in the routing table, a router that connects the network may not be passing packets because of filters that may have been set up; if the network you are trying to reach is not in the routing table, it is unreachable. From the table entries, or lack of table entries, you can determine which path is being used and in what direction you can continue to investigate.

- 6 Use the APING and ANameLookup commands to determine the connectivity to different router and end stations.

You can determine where the connectivity is broken by how far you can see. See your network planning documentation for the intended connectivity. For a detailed description of the APING and ANameLookup commands, see the Commands chapter in *Reference for Enterprise OS Software*.

Unless you have fully meshed Frame Relay or X.25 AppleTalk network topologies, the APING and ANameLookup commands may not work with router ports attached to these wide area network media. It is recommended that you use the APING command against AppleTalk local area network ports on these routers to determine reachability.

- 7 If your router has a serial line interface, check the transmit clock to see if it is correctly set using:

```
SHow !<path> -PATH CLock
```

- 8 Check that all cables on all routers in a specific path in the routing table are properly connected and that the routers are properly installed.

For instructions, see the installation guide provided with your bridge/router.

- 9 Check AppleTalk statistics by entering:

```
SHow -SYS STATistics -AppleTalk
```

For complete information on AppleTalk statistics, see the Statistics Displays appendix.

Customizing the AppleTalk Router

Most AppleTalk parameters are automatically configured to their default values. (With few exceptions, the only parameters that need to be configured to enable routing are discussed in "Setting Up a Basic AppleTalk Router" earlier in this chapter.) In some cases, you may want to change the default configuration.

This section is intended for those who want to go beyond the minimum configuration of a nonseed or seed router. It explains how to:

- Set up AppleTalk routing over a non-AppleTalk data link.
- Change the frequency at which a routing table propagates routes.
- Set up filters.
- Change a zone list for an AppleTalk network.

Not all available parameters are discussed in this section. For more information on all available parameters, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

Setting Up Multiple Seed Routers

This section provides information on setting up multiple seed routers on a network.

Procedure

To install multiple seed routers on a network, see "Setting Up a Basic AppleTalk Router" earlier in this chapter.

Related Information

To provide redundancy in case of system crashes and power outages, you can install multiple seed routers on the same network.

When you install more than one 3Com AppleTalk router as a seed router for a particular network, all the routers should seed the same information configured for that network. The first seed router that establishes itself (is started and goes active) on the network becomes the actual seed router. After one or more AppleTalk routers are started up, the seed information provided by the seed router can be supplied by any of the routers connected to a particular network.

To display any network number inconsistencies between routers, enter:

```
SHow -AppleTalk DIAGnostics
```

The first seed router that establishes itself on a network defines the values. The subsequent NETBuilder seed routers discovering the inconsistency can optionally, if the SeedCheck option is selected (default setting), disable the port connected to the network and note the condition that is displayed.



Different brands of AppleTalk routers handle conflicting seed information differently. For details of their operation, see their respective documentation.

Setting Up AppleTalk Routing over a Non-AppleTalk Data Link

To configure a local or wide area port of a router connected to a non-AppleTalk data link, follow these steps:

- 1 Enable AppleTalk routing over a non-AppleTalk network using:

```
SETDefault !<port> -AppleTalk CONTROL = (RoutE, NonAppleTalk)
```

- 2 Verify the configuration of each router port using:

```
SHow !<port> -AppleTalk CONFIGuration
SHowDefault !<port> -AppleTalk CONFIGuration
```

The SHow configuration command displays live values. The SHowDefault command displays the values you have configured.

Related Information

Where AppleTalk routing is supported, any data type such as Ethernet, FDDI, token ring, PPP, X.25, SMDS, or Frame Relay can be treated as a non-AppleTalk link, backbone, or "cloud." 3Com AppleTalk routers can communicate across these links, connecting the AppleTalk networks that exist as offshoots of the data link.

This feature is especially useful for configuring the point-to-point links (PPP) and cloud links (X.25, Frame Relay), where no AppleTalk end systems can reside. Although any of the remaining data links (Ethernet, token ring, FDDI, SMDS) can support AppleTalk end nodes, they may not support them in actual installations. They may operate as a backbone network, or only support non-AppleTalk network devices.

When AppleTalk end nodes are not supported, if you configure the links as non-AppleTalk, you do not need to configure seed information, which saves network range numbers and zone lists. Unwanted name lookup multicasts on the

link are also eliminated (most commonly generated by using the Chooser interface on the Macintosh).

A disadvantage to configuring Frame Relay and X.25 ports when connected to non-AppleTalk networks is the work involved in moving configured neighboring router information to another port. If you move a serial interface to a different port, you need to define the neighbor information for the new port and delete the same information from the old port using the `-AppleTalk ADDRess` parameter. If you treat the port as connected to an AppleTalk network, you only need to define the network range on the new port and remove the same range from the old port. (To define and delete the network range, use the `SETDefault -AppleTalk NetRange` command.) The software automatically associates the configured neighbor information (for example, `20.30 @56`) with the new port when it is activated.

Changing Frequency of Routing Table Route Propagation

This section provides information on how to change the frequency at which a routing table propagates routes.

Procedure

To change the frequency, follow these steps:

- 1 Change the frequency at which a router sends out routing information packets using:

```
SETDefault -AppleTalk RouteUpdateTime = <seconds> (1-300)
```

- 2 Change the frequency at which routes in the routing table are verified using:

```
SETDefault -AppleTalk RouteAgingTime = <seconds> (20-300)
```

Related Information

Every 10 seconds (the default setting of the `RouteUpdateTime` parameter), the router sends broadcast packets to its neighboring routers to report the following types of information:

- The networks it can reach
- The number of hops associated with each network it can reach

You can configure the `RouteUpdateTime` parameter to change the frequency at which the router sends out routing information packets.

When other AppleTalk routers that cannot change the time interval are present do not use a value other than the default of 10 seconds. The value of the `RouteUpdateTime` parameter and the frequency of AppleTalk routing table aging are related. Table aging is set through the `RouteAgingTime` parameter, which has a default of 20 seconds. If broadcasts are less frequent, but aging is left the same or reduced, increased table entry deletions and additions may occur, which can affect routing capability and increase table maintenance overhead.

Try to keep at least a 1-to-2 ratio between `RouteUpdateTime` and `RouteAgingTime`. However, increasing the value of both parameters increases the time for topological changes to propagate through the routers. Route update packets also are not reliably received and may be lost on a busy network. Their frequency should be enough to ensure reception on a busy network before other routers age out the routes. Decreasing the value of both parameters improves

route propagation and route convergence to new paths, but at the expense of higher bandwidth utilization for route information exchange.

When a route is learned, it goes into the routing table. The router then sends a query asking for zone lists for the networks for which it does not have complete zone list information. Other routers pass back zone list information to the querying router. This occurs as information about other networks are propagated. A NETBuilder II router does not propagate information about route information for a network until it has complete zone list information associated with that network.

Setting Up Filters

The following types of filtering are available for restricting access to the AppleTalk internetwork through a specified port:

- Network number-based filtering
- Entity filtering

The use of both network and entity filters is only effective if there are no alternate, non-filtered routes to the filtered networks or services. The use of filtering also slows down the performance of your AppleTalk router.

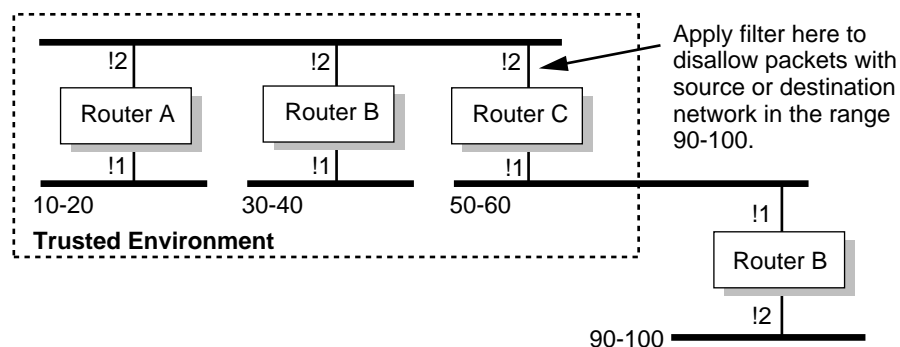
The following sections describe each type of filtering. For more examples and details on using the parameters described in these sections, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

Setting Up Network Number-Based Filtering

This section provides procedures on how to set up positive and negative network number-based filtering. A sample topology is also provided to illustrate each step of the procedures.

The following example is an application of network-number based filtering. In this example, three AppleTalk networks are interconnected through a backbone (networks 10–20, 30–40, 50–60 in Figure 208). These three networks are said to be in a “trusted” environment; that is, nodes on these networks can access resources on all three networks. A second network (90–100 in Figure 208) is said to be outside the trusted environment. Nodes on that network are permitted to access resources on network 50–60 (and vice versa) but are prevented from accessing resources on the other networks connected to the backbone (namely 10–20 and 30–40).

Figure 208 AppleTalk Network Filter Example



One way you can satisfy these requirements is with *positive filtering*. As shown in Figure 208, this filtering is implemented by applying network filtering of 90–100 on port 2 of router C. This filter stops the propagation of packets either originating from a node or destined to a node with a network number in the range 90–100 beyond this interface. In other words, if a packet from a node on network 90 is received on port 1 of router C and is destined to a node on network 10, then it is not forwarded out of port 2 of router C. Similarly, if a packet is received from a node on network 10 on port 2 and is destined to a node on network 90, it is not forwarded out of port 1.

Setting Up Positive Filtering To set up a positive network filter, follow these steps. The sample topology described previously will be used to illustrate each step.

- 1 Enable network number filtering using:

```
SETDefault !<port> -AppleTalk CONTROL = NetFilter
```

- 2 Create a set of filter network ranges using:

```
ADD !<port> -AppleTalk NetFilter = <network range>
```

- 3 Specify that the newly created network filter range is to be used for positive filtering using:

```
SETDefault !<port> -AppleTalk NetFilterType = Positive
```

In the previous procedure on how to set up positive filtering, the filtered set of networks is included within the specific range of 90–100. You can achieve the same results with *negative filtering*, which is the application of filtering through exclusion. In this case, the filtered set of networks are all networks *not* in the range 10–60. You can apply this filter at the same point, that is, port 2 of router C.

Setting Up Negative Filtering The sample topology described above will be used to illustrate each step of the following procedure.

To set up negative filtering, follow these steps:

- 1 Enable network number filtering using:

```
SETDefault !<port> -AppleTalk CONTROL = NetFilter
```

- 2 Create a set of filter network ranges using:

```
ADD !<port> -AppleTalk NetFilter = <network range>
```

- 3 Specify that the newly created network filter range is to be used for negative filtering.

For example, to set the network filter range specified on port 2 of router C to positive, enter:

```
SETDefault !2 -AppleTalk NetFilterType = Negative
```

For complete information on each of the parameters used in this section, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

Related Information Network filtering allows you to filter received packets on a per-port basis based on source and destination network numbers. The following criteria apply:

- Packets are filtered on receipt at a port based on a packet's final destination network.
- Packets are filtered on forwarding (transmission) out of a port based on the network from which the packet originated.

These criteria control the flow of packets between the various ports of a router. The following events also occur as a result of filtering:

- Networks are not included in Routing Table Maintenance Protocol (RTMP) routing updates out a port if their range is completely included in the set of filtered networks for the port.
- Zone information is suppressed from being sent out a port if all networks associated with a zone are in the set of filtered networks for the port and the zone is not associated with the directly connected network out the port.

The following types of network filtering are available:

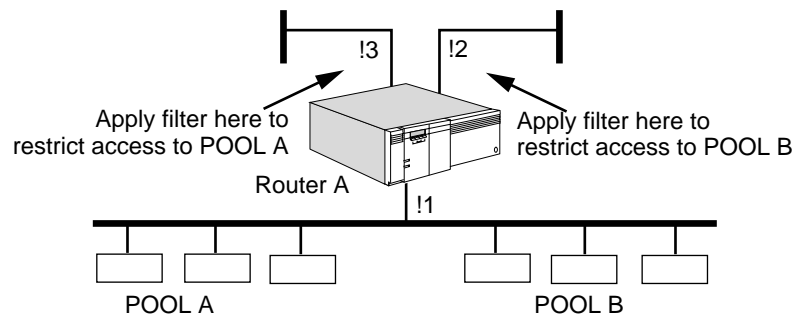
- Positive network filters discard all packets destined to or originating from a set of network number ranges that you specify.
- Negative network filters discard all packets except for those destined to or originating from a set of network number ranges that you specify.

Setting Up Entity Filters

This section provides a procedure on how to set up entity filtering. A sample topology is provided to illustrate each step of the procedure. At the end of the procedure, an additional example of implementing entity filtering is provided.

In Figure 209, router A has three ports. Port 1 is connected to a network that contains two pools of resources, labeled POOL-A and POOL-B. These resources could be a collection of printers, file servers, communication servers, etc. Port 2 and port 3 are connected to two network segments that contain users who access the resources in POOL-A and POOL-B. The requirement in this example is to partition the pool of resources so that all users on the segment attached to port 2 can only access resources in POOL-A and all users on the segment attached to port 3 can only access resources in POOL-B. To simplify the filter specification, assume that all resources in POOL-A have object names with the prefix "POOL-A" and all resources in POOL-B have object names with the prefix "POOL-B," for example, "POOL-A-LASERWRITER," and "POOL-B-DBSERVER,."

Figure 209 AppleTalk Entity Filter



As shown in Figure 209, the entity filters are applied at ports 2 and 3. At port 2, the filtered set will be all entities whose object names start with the pattern "POOL-B." At port 3, the filtered set will be all entities whose object names start with the pattern "POOL-A." The configuration of these filters is shown in the following procedure.

Procedure To set up your AppleTalk router to perform entity filtering, see Figure 209 and follow these steps :

1 Enable entity filtering.

In the topology shown in Figure 209, entity filtering should be enabled on ports 2 and 3. For example, to enable entity filtering on port 2, enter:

```
SETDefault !2 -AppleTalk CONTROL = EntityFilter
```

2 Create one or more entity filters.

Create entity filter specification "POOL-A~::~=@=" and make it filter number 1 in the entity filter table by entering:

```
ADD -AppleTalk EntityFilter 1 "POOL-A~::~=@="
```

Create entity filter specification "POOL-B~::~=@=" and make it filter number 2 in the entity filter table by entering:

```
ADD -AppleTalk EntityFilter 2 "POOL-B~::~=@="
```

3 Assign an entity filter to a particular port and specify whether it is a positive or negative filter.

To assign entity filter number 1 to port 3 and specify that it is a positive filter, enter:

```
ADD !3 -AppleTalk EntityFilterNum 1 Positive
```

The statistic Entity Filter Matches is present at the end of the AppleTalk statistics. It displays the number of NBP Request or Reply packets dropped because of a match against an active entity filter.

Assign entity filter number 2 to port 2 and specify that it is a positive filter by entering:

```
ADD !2 -AppleTalk EntityFilterNum 2 Positive
```

Example To create an entity filter that restricts access to a LaserWriter with the name "MktPrinter" in zone "Mkt," enter:

```
ADD -AppleTalk EntityFilter 1 "MktPrinter:LaserWriter@Mkt"
```

To define that the above entity filter is a positive filter that applies to port 2, enter:

```
ADD !2 -AppleTalk EntityFilterNum 1 Positive
```

For complete information on each of the parameters used in this section and more examples on how to create entity filters, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

Related Information Entity filtering allows you to restrict access across a port to specific named network entities or sets of entities on an AppleTalk network. These resources can include file servers, printers, and communications servers. Access to network entities is based on entity name and (optionally) network number.

Entity filtering operates as a filter on name lookup requests and responses across a port. When a Macintosh user opens the Chooser interface and selects a service icon, name lookups are sent across the internetwork to all networks that are associated with the zone currently selected in the Chooser. Those services that meet the lookup criteria (in this case, those that have the same entity name type in the specified zone) send lookup response packets containing their entity name

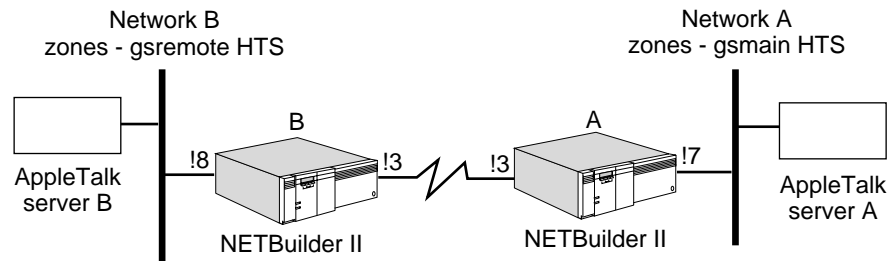
back to the source of the lookup. Entity filtering prevents responses from being returned by stopping the requests from continuing on or by intercepting the responses. It checks in both directions because wildcards are used in the requests, but not in the responses.

The configuration of entity filters is a two-step process. The first step is to configure filtering criteria by specifying the entity name and, optionally, a network number range qualifier. This information is configured through the EntityFilter parameter. The second step is to associate filtering criteria with a port in addition to the positive or negative filter type attribute. You can use the EntityFilterNum parameter to add an entity filter to a specified port and designate it as a positive or negative filter.

Setting Up Zone Advertisement Filtering

This section provides a procedure on how to set up zone advertisement filtering. A sample topology is provided to illustrate each step of the procedure.

Figure 210 Setting Up Zone Advertisement Filtering



Zone advertisement filtering filters specific zones being returned in the ZoneList when a ZIP ZoneList request is received. A Zip ZoneList is created when:

- A station is connected to the AppleTalk network and it is acquiring the available zones.
- A chooser application is acquiring zones to access shared devices.
- An InterPool application is requiring zones to list all devices.

The zone advertisement filter is configured on specific ports, which allows zones to be hidden on some ports but advertised on others. Zones are configured using the EntityFilter parameter in the AppleTalk Service. Only zone-specific filters (" :=@zone") can be selected for a zone advertisement filter. This procedure is demonstrated in the following example.

An AppleTalk network consists of two networks connected through a serial port. Each network contains its own private zone ("gsremote" and "gsmain") and one common zone ("HTS"). The user wants to allow the resources on the common zone to be accessible by both networks but keep the resources on the private zone accessible only by the local network.

Procedure

To configure zone advertisement filtering on NETBuilder II A, follow these steps:

- 1 Configure entity filter " =:@HTS" and assign the entity filter number 1 by entering:

```
ADD -AppleTalk EntityFilter 1 " =:@HTS"
```

- 2 Configure entity filter number 1 to port 7 by entering:

```
ADD !7 -AppleTalk ZoneAdvFilterNm 1 Negative
```

- 3 Enable zone advertisement filtering on port 7 by entering:

```
SETD !7 -AppleTalk CONTROL = ZoneAdvFilter
```

From network A, only zones " HTS" and " gsmain" will be advertised. The zone " gsmain" is advertised because it is the local zone for network A. A Chooser or Interpool will only see those two zones.

From network B, zones " HTS", " gsmain", and " gsremote" will be advertised. There are no zone advertisement filters configured on NETBuilder II B. A Chooser or Interpool on network B will see all the zones.

- 4 Prevent gsmain from being advertised to network B port 8 by entering:

```
ADD -AppleTalk EntityFilter 1 " =:@HTS"
```

```
ADD !8 -AppleTalk ZoneAdvFilterNum 1 Negative
```

```
SETD !8 -AppleTalk CONTROL = ZoneAdvFilter
```

A chooser or Interpool will only see zones " HTS" and " gsremote" The zone " gsremote" was advertised because it is the local zone for network B.

Procedure

To use per-port directional entity filtering to achieve the same effect as zone advertisement filtering, follow these steps:

- 1 Configure per-port entity filtering on NETBuilder II A by entering:

```
ADD -AppleTalk EntityFilter 1 " =:@HTS"
```

- 2 Configure entity filter number 1 to port 7 by entering:

```
ADD !7 -AppleTalk EntityFilterNum 1 Negative ClientIn
```

The negative value specifies that only NBP Requests (" =:@HTS") entering port 7 will be allowed.

- 3 Enable entity filtering on port 7 by entering:

```
SETD !7 AT CONTROL = EntityFilter
```

The Zip ZoneList request will return all the zones. Therefore, the Chooser or Interpool will see zones " HTS," " gsmain," and " gsremote." When the Chooser or Interpool tries to find devices on " gsremote," the NBP request will be filtered.

- 4 To prevent a Chooser or Interpool from network B from accessing " gsmain" devices, set the entity filter to filter NBP requests exiting port 7 by entering:

```
DELEte !7 -AppleTalk EntityFilterNum 1
```

```
ADD !7 -AppleTalk EntityFilterNum 1 Negative ClientBoth
```

The ClientBoth parameter applied the filter to both NBP requests entering and exiting port 7. The negative value specifies that only NBP requests (" =:@HTS") are allowed.

Both configurations can be done to filter zone advertisements and NBP requests simultaneously.

Changing a Zone List

You may need to change a zone list on an AppleTalk network to add a new subset of devices for service access on a large link or to correct an error introduced during the initial configuration.

To change a zone list, follow these steps:

- 1 Disable AppleTalk routing on all ports (on all routers) connected to the AppleTalk network using:

```
SETDefault !<port> -AppleTalk CONTROL = NoRoute
```

- 2 Reconfigure all seed routers on the AppleTalk network with the same zone list and default zone.

- To add a zone name to the zone list, use:

```
ADD !<port> -AppleTalk ZONE "<zone-string>" (1-32 char)
```

- To delete a zone name from the zone list, use:

```
DELETE !<port> -AppleTalk ZONE "<zone-string>" (1-32 char)
```

- To update the default zone, use:

```
SETDefault !<port> -AppleTalk DefaultZone = "<zone-string>" (1-32 char)
```

- 3 After all routers on the extended AppleTalk internetwork have aged out the network from their routing tables, re-enable AppleTalk routing on all ports that you disabled earlier in step 1 using:

```
SETDefault !<port> -AppleTalk CONTROL = Route
```



A 15-minute wait is adequate for large networks. On some networks, re-enabling the routers too soon may result in difficulty in determining the zones for a specific network or finding services because some routers may have different zone lists for the same network.

3Com recommends rebooting all other AppleTalk devices on the modified AppleTalk internetwork, although some devices may adjust more easily. If there are any problems, reboot the router.

How the AppleTalk Router Works

This section discusses AppleTalk routing concepts, including information about using seed routers to provide network numbers and zone names to a connected network.

3Com bridge/routers provide complete AppleTalk Phase 2 routing capability by broadcasting routing information, forwarding packets, and responding to routing-related requests from AppleTalk-based workstations and other routers.

An AppleTalk router identifies information (including network numbers and zone names) for directly connected AppleTalk networks. The router uses network numbers to determine how to forward data to other networks on the AppleTalk internetwork. The router keeps zone information, which divides the internetwork into logical subdivisions, to help users access services through the AppleTalk internetwork.

Each of the ports associated with a physical interface on the system is considered to be connected to a different network. You determine which network ports on the system support AppleTalk routing.

Any grouping of networks connected by AppleTalk routers is known as an AppleTalk internetwork; each network on an internetwork can be on different physical media (for example, Ethernet, token ring, and FDDI).

The router that contains the primary identifying information associated with a physical network is called a *seed router*. A seed router must be the first router to be brought up on a network, preferably before any other AppleTalk devices are booted on the network. If a router is not a seed router for a network, it obtains the identifying information for the network (the network range, associated zone list, and default zone) from a seed router that is attached to the same network. After a router acquires the seed information from the seed router, it also can provide seed information to other routers and end nodes subsequently activated on the same network.



If bridging is enabled, AppleTalk Phase 1 packets are bridged through all active interfaces, regardless of the state of AppleTalk Phase 2 routing.

The identifying information that an AppleTalk Phase 2 router uses to keep track of networks on the internetwork includes:

- A network number range associated with each network.
- A zone list associated with each network.

A network number range is a unique range of contiguous network numbers, for example, 110–120, that identifies a particular AppleTalk network in a Phase 2 internetwork. A LocalTalk network, sometimes referred to as a non-extended network, is always identified by a network range consisting of a single network number (for example, 30–30) and a single associated zone. A network number in AppleTalk Phase 2 can be any number from 1 to 65,279 (0001 to hex FFFF).

The AppleTalk network number is the portion of packet destination addresses that allows the router to identify and route AppleTalk packets to the correct network.

A zone groups AppleTalk devices (nodes) within one or more networks so users can easily locate and access services (for example, printers and file servers). The networks or devices within a zone do not have to be adjacent or share common routers. Typically, they are geographically adjacent for routing efficiency and easy physical access to devices, such as printers.

The number of zone names you associate with a network depends on the size of the internetwork you are planning. If your internetwork is small, a single zone name may be adequate for all networks. If a single Ethernet or token ring network spans a large geographic area or contains large numbers of AppleTalk devices (such as printers or file servers), then use multiple zones to make it manageable for users.

In AppleTalk Phase 2, LocalTalk networks must be associated with a single zone; Ethernet, token ring, FDDI, and SMDS networks can be associated with multiple zones. In AppleTalk Phase 2, a default zone is identified within the zone list for a network; the default zone is defined by a seed router. Individual nodes on a network are usually automatically configured to be in the default zone, and can be explicitly configured to be in a different zone present in the network's zone list.

AppleTalk routers also use the mapping of zones to networks to support the distributed name database maintained by the AppleTalk Name Binding Protocol (NBP).

The Apple Macintosh Chooser interface provides the most common point of exposure to zones. If two or more zones exist on the connected AppleTalk internetwork, a list of all zones across all networks is presented to the user. When a user selects a zone and service icon in the Macintosh Chooser, the user sees a list of only those services that exist in the zone. For example, instead of selecting from a list of 20 LaserWriter printers connected to an internetwork, a user may see only the two LaserWriter printers that are within the selected zone. This feature makes printer and other service selection both easier and faster.

Macintosh users can determine, through the icons within the Network Control Panel, what zone they will default to; this choice is reflected in the initial zone that appears in the Chooser interface.

To display network-to-zone mapping information, enter:

```
SHow -AppleTalk NetZoneMapping
```

To list all networks that are associated with each zone in the AppleTalk internetwork, enter:

```
SHow -AppleTalk ZoneNetMapping
```

For information on assigning zone names and other zone-related functions, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

Network Entities

A network entity is a named AppleTalk entity, usually a service (such as file service or a printer) associated with an AppleTalk socket on an AppleTalk node.

The entity name is a character string enclosed in quotes and made up of three fields: object, type, and zone. Object and type are separated by a colon; type and zone are separated by the at (@) sign. Up to 32 characters are allowed for each field in the entity name. Entity names are case-insensitive. The following is an example of an entity name:

```
"AppleShare Server:AFPServer@engineering"
```

If you are familiar with the Macintosh Chooser, the object name of network entities appears in the upper right corner. The type is a name associated with the icons that appear in the upper left corner, but not necessarily the same as the name under the icon itself. The zone, if more than one zone is defined in the AppleTalk internetwork, will be in a zone list in the bottom left corner.

Within the bridge/router, network entity names perform the following tasks:

- Name router ports for discovery and APING. (For information on APING, see the Commands chapter in *Reference for Enterprise OS Software*.)
- Describe entity filter specifications. (For more information on entity filtering, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.)

Object and zone strings are names that appear in the AppleTalk network-aware user interface, primarily with the Macintosh Chooser window, but also in third-party applications. The character set used in these strings is the extended ASCII character set used within the Macintosh. The bridge/router user interface for

AppleTalk provides a universal representation of the extended AppleTalk ASCII character set. The extended character set permits the use of foreign characters in configured strings (for example, zone names) that are seen by AppleTalk end systems. Foreign language characters can also be entered as input to query functions (for example, ANameLookup command), and names containing such characters can be displayed without loss of information.

To enter these characters, key an escape character followed by a two-digit hex code for the desired character. For example, to enter an ñ, you first enter the escape character, which is a backslash (\), followed by the two-character hex code for the desired character as listed in Table 51. In this case, the hex code is 96 (an ASCII value (decimal) of 150). To specify a zone string such as "mañana," you would enter the eight-character string "\ma\96ana."

The AppleTalk Service displays the string in the same format in which it is obtained from another AppleTalk device. On a Macintosh, the example would appear as "mañana" in the Chooser, assuming that multiple zones are defined within the AppleTalk internetwork.

Table 51 Macintosh Extended Character Set

| ASCII Value | Hex Equivalent | Macintosh Character | ASCII Value | Hex Equivalent | Macintosh Character |
|-------------|----------------|---------------------|-------------|----------------|---------------------|
| 128 | 80 | À | 159 | 9F | Ü |
| 129 | 81 | Á | 160 | A0 | † |
| 130 | 82 | Ç | 161 | A1 | ° |
| 131 | 83 | É | 162 | A2 | ¢ |
| 132 | 84 | Ñ | 163 | A3 | £ |
| 133 | 85 | Ö | 164 | A4 | § |
| 134 | 86 | Ü | 165 | A5 | • |
| 135 | 87 | á | 166 | A6 | ¶ |
| 136 | 88 | à | 167 | A7 | ß |
| 137 | 89 | â | 168 | A8 | ® |
| 138 | 8A | ä | 169 | A9 | © |
| 139 | 8B | ã | 170 | AA | ™ |
| 140 | 8C | â | 171 | AB | ' |
| 141 | 8D | ç | 172 | AC | " |
| 142 | 8E | é | 173 | AD | ≠ |
| 143 | 8F | è | 174 | AE | Æ |
| 144 | 90 | ê | 175 | AF | Ø |
| 145 | 91 | ë | 176 | B0 | ∞ |
| 146 | 92 | í | 177 | B1 | ± |
| 147 | 93 | ì | 178 | B2 | ≤ |
| 148 | 94 | î | 179 | B3 | ≥ |
| 149 | 95 | ï | 180 | B4 | ¥ |
| 150 | 96 | ñ | 181 | B5 | μ |
| 151 | 97 | ó | 182 | B6 | ð |

Table 51 Macintosh Extended Character Set (continued)

| ASCII Value | Hex Equivalent | Macintosh Character | ASCII Value | Hex Equivalent | Macintosh Character |
|-------------|----------------|---------------------|-------------|----------------|---------------------|
| 152 | 98 | ò | 183 | B7 | Σ |
| 153 | 99 | ô | 184 | B8 | Π |
| 154 | 9A | ö | 185 | B9 | π |
| 155 | 9B | õ | 186 | BA | ∫ |
| 156 | 9C | ú | 187 | BB | ª |
| 157 | 9D | ù | 188 | BC | º |
| 158 | 9E | û | 189 | BD | Ω |
| 190 | BE | æ | 223 | DF | fl |
| 191 | BF | ø | 224 | E0 | ‡ |
| 192 | C0 | ¿ | 225 | E1 | · |
| 193 | C1 | ¡ | 226 | E2 | , |
| 194 | C2 | ¬ | 227 | E3 | „ |
| 195 | C3 | √ | 228 | E4 | ‰ |
| 196 | C4 | f | 229 | E5 | Ã |
| 197 | C5 | ≈ | 230 | E6 | Ê |
| 198 | C6 | Δ | 231 | E7 | Á |
| 199 | C7 | « | 232 | E8 | È |
| (continued) | | | | | |
| 200 | C8 | » | 233 | E9 | Ë |
| 201 | C9 | ... | 234 | EA | Í |
| 202 | CA | | 235 | EB | Î |
| 203 | CB | À | 236 | EC | Ï |
| 204 | CC | Ã | 237 | ED | Ì |
| 205 | CD | Õ | 238 | EE | Ó |
| 206 | CE | Œ | 239 | EF | Ô |
| 207 | CF | œ | 240 | F0 | 🍏 |
| 208 | D0 | – | 241 | F1 | Ò |
| 209 | D1 | — | 242 | F2 | Ú |
| 210 | D2 | “ | 243 | F3 | Û |
| 211 | D3 | ” | 244 | F4 | Ü |
| 212 | D4 | ’ | 245 | F5 | ı |
| 213 | D5 | ’ | 246 | F6 | ˆ |
| 214 | D6 | ÷ | 247 | F7 | - |
| 215 | D7 | ◊ | 248 | F8 | - |
| 216 | D8 | ÿ | 249 | F9 | ˘ |
| 217 | D9 | ÿ | 250 | FA | ˙ |
| 218 | DA | / | 251 | FB | ˚ |
| 219 | DB | | 252 | FC | ˛ |

Table 51 Macintosh Extended Character Set (continued)

| ASCII Value | Hex Equivalent | Macintosh Character | ASCII Value | Hex Equivalent | Macintosh Character |
|-------------|----------------|---------------------|-------------|----------------|---------------------|
| 220 | DC | < | 253 | FD | ~ |
| 221 | DD | > | 254 | FE | . |
| 222 | DE | fi | 255 | FF | ˘ |

Port Startup Operations

After you set up and check the router according to the instructions in the previous sections, it is ready to do some packet routing. The following actions occur when the AppleTalk router (with `CONTRol` set to `ROute` and `AppleTalk`) starts up on a port connected to an AppleTalk network:

- The router acquires a provisional AppleTalk node address for the port using AppleTalk Address Resolution Protocol (AARP) until the final network range for the connected network is known. (Frame Relay, X.25, and PPP must be statically configured with a final address.)
- If the `CONTRol` parameter is set to `SeedingAllowed`, and the seed information is configured using the `NetRange`, `ZONe`, and `DefaultZone` parameters, the following applies:
 - Using AARP, the router dynamically acquires a final AppleTalk node address with the network number taken from the configured network range. If the value of the `StartupNET` parameter is within the configured network range, the values for the `StartupNODE` (if nonzero) and `StartupNET` parameters are used as first attempt values in the process. If these values are tried but are already in use by another node, then an attempt is made to use the last address acquired from the previous startup, provided that it is in the proper network range. If this also fails, then the router finds a unique address in the configured network range.
 - If `SeedCheck` is enabled, and locally configured seed information is different from that seen for any other router on the network during the first twenty seconds of port activity, then the port is disabled. Information describing conflicting configurations is saved. You can display the information that describes configuration conflicts using the `SHow -AppleTalk DIAGnostics` command.
 - If `NoSeedCheck` is enabled, the router uses the locally configured seed information. If a difference in seed information between the local configuration and any other router on the connected network is detected, the last occurrence of conflicting information detected is saved. A difference in seed information does not disable the port in this case. You can display the conflicting information using the `SHow -AppleTalk DIAGnostics` command.
- If the `CONTRol` parameter is set to `NoSeedingAllowed`, or you do not have sufficient seed information configured, then the router does not seed, but waits for a seed router to appear.
 - If a seed router appears on the connected network, the router obtains the seed information from that router and proceeds.

The router performs dynamic node address acquisition using AARP by selecting a network number from the network range given in the seed information. If the value of the `StartupNET` parameter is within the

configured network range, then the values for the StartupNODE (if nonzero) and StartupNET parameters are used as first attempt values in the process. If these values are tried but are already in use by another node, then an attempt is made to use the last address acquired. If this also fails, then the router finds any unique address in the configured network range.

- If no other seed router is detected on the connected network, the router remains in this listening state indefinitely.

After seed information is established for at least one of the active ports, the router begins to construct a routing table, which contains next router and distance information for all reachable networks and zone lists for each network. The tables are constructed from routing information (RTMP) packets received periodically from other routers. As new routes are discovered from these packets, the receiving router will ask the sending router for zone list information for each new network.

The maintenance of zone list information by the router allows the router to support access by AppleTalk end systems to named network entities. Routers supply AppleTalk end systems with the list of zones to assist in the location of end services. AppleTalk routers also support the discovery of named entities by using zone-to-network associations present in the routing tables.

Network AppleTalk Operations

This section provides an overview of AppleTalk operations on the network, particularly the routing function.

An AppleTalk network is usually configured on each port where AppleTalk packets are received and sent. The port can be a local area port, such as Ethernet, FDDI, or token ring. SMDS is supported over extended distances in an almost identical manner to that of the local area networks. AppleTalk can also be routed over a backbone network not configured as an AppleTalk network. This routing is usually done with a serial line port for a wide area network, such as a PPP, X.25, SMDS, or Frame Relay link. PPP, X.25, and Frame Relay links can also be set up as AppleTalk networks, but more configuration is required.

A router must check its routing table to determine where to route a packet. If the destination node is on a directly connected network, the router sends the packet directly to the destination node. If the network identified in the destination address is not directly connected, the packet is forwarded to the next router in the route to the destination network as maintained in the routing table.

For an example and description of the AppleTalk routing table, see the AllRoutes parameter in the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

Figure 211 is an example of an AppleTalk internetwork. The upper router depicted in the Engineering Zone is a seed router on port 1. The following zone information should be configured on the indicated routers to provide the pictured zone boundaries:

Finance Zone router port 1: Zonelist: Finance
port 2: PortZone: Finance

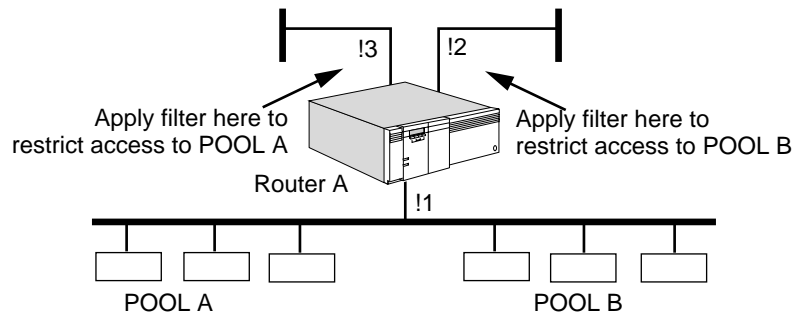
Marketing Zone router port 1: can take defaults from a seed router
PortZone: Marketing

port 2: Zonelist: Marketing

Engineering Zone upper router port 1: Zonelist: Finance, Engineering, Marketing
port 1: Default Zone: Marketing (presumably most end nodes are in Marketing)
port 1: PortZone: Engineering

Engineering Zone lower router port 1: Zonelist: Engineering

Figure 211 AppleTalk Network



Split Horizon The AppleTalk router uses the split horizon routing method. This routing method helps reduce network traffic by not broadcasting route information for a network out the same interface over which the network's route was learned.

For Frame Relay and X.25 ports, split horizon decisions are made at the next router link level instead of at the port level. This feature allows support for nonmeshed topologies by allowing a router to use a Frame Relay or X.25 port as a virtual hub, sending route information to each router out the port learned from all other routers out of the same port. If the decisions were made at the port level, as is the case for AppleTalk on LANs and SMDS, no routing information learned from any router out of the port would be sent to any router out of the same port.

AppleTalk over PPP A PPP link routing AppleTalk is normally configured as a non-AppleTalk data link because PPP does not support AppleTalk Address Resolution Protocol (AARP). The two sides of the link may choose the same network and node address if the link is configured as an AppleTalk data link. In this case, AppleTalk routes are not updated properly on both sides of the link. If you decide to configure a PPP link as an AppleTalk data link, enter unique startup network but different unique startup node numbers on the PPP port of both routers using:

```
SETDefault !<port> -AppleTalk StartupNET = <number> (0-65279)
SETDefault !<port> -AppleTalk StartupNODE = <number> (0-253)
```

Filtering on Frame Relay Ports To apply filtering to or from specific neighbors out of the same Frame Relay port, you must use the virtual port feature. For more information on virtual ports, see the Configuring Advanced Ports and Paths chapter.

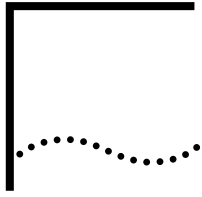
Routing Table Access the AppleTalk routing table using:

```
SHow !<port> -AppleTalk AllRoutes
```

For a sample display and explanation of a routing table, see the AppleTalk Service Parameters chapter in *Reference for Enterprise OS Software*.

The RTMP establishes and maintains the AppleTalk routing tables. Routing table entries identify the shortest possible path (measured in hop counts) to the network by identifying the next route to which packets should be sent.

AppleTalk always selects the route that requires the fewest hops. When packets are forwarded, a hop count field is incremented. Packets with a hop count of 15 or more are not forwarded to avoid indefinite looping.



CONFIGURING DECNET ROUTING

This chapter describes the procedures for configuring your system to perform DECnet Phase IV routing, route filtering, Phase IV to Phase V transition, and internetworking. It also describes how the router works and gives guidelines for operating, managing, and troubleshooting it.



For conceptual information, see “How the DECnet Router Works” later in this chapter. If you need help with terminology, see “DECnet Phase V and Phase IV Terms” later in this chapter.

Setting Up a Basic DECnet Router

The procedures in this section describe how to route DECnet packets within a DECnet network. Depending on your network requirements, you can use the default values of the parameters in the DECnet Service, or you may want to further configure the router according to “Customizing the Configuration” later in this chapter.

DECnet routing supports multiple independent DECnet networks attached to the router. It also allows internetwork routing either among all nodes on the selected networks, or between specific nodes on selected networks through user-defined address translations.



Unless otherwise noted, each command in the following procedures can be used whether you are configuring a Level 1 (intra-area) or Level 2 (interarea) router.

Configuring for Local Area Networks and Point-to-Point Links

Use this procedure to configure basic DECnet routing over LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Log on to the system with Network Manager privilege and set up the ports and paths of your bridge/router according to the procedure in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Procedure

To configure the bridge/router to perform DECnet routing, follow these steps:

- 1 Set the DECnet address for this router using:

```
SETDefault -DECnet ADDRESS = None | <area number>.<node  
number>(1-63).(1-1023) [<network> (0-15)]
```

The area number is a decimal number in the range of 1 to 63. The node number is a decimal number in the range of 1 to the value specified for the MaxNodeNumber parameter. The node number must be unique within an area number. For example, if a node with the address 3.1 already exists, do not set the address for this router to 3.1.

The value entered for the area number should not exceed the value configured for the MaxAreaNumber parameter. The default for the area number is 63. The value entered for the node number should not exceed the value entered for the MaxNodeNumber parameter. The default for the maximum node number is 255. For more information on the MaxAreaNumber and MaxNodeNumber parameters, see the DECnet Service Parameters chapter in *Reference for Enterprise OS Software*.



Enable bridging and all routing protocols before enabling DECnet.

- 2 Enable DECnet routing on a particular port using:

```
SETDefault !<port> -DECnet CONTrol = ROute
```

Repeat this step for other ports, including serial line ports.

If DECnet routing is enabled on a serial line port, the system at the other end of the serial line also must be routing (not bridging) DECnet traffic. DECnet assumes that serial lines are point-to-point links. Bridging DECnet packets on the other end of the serial line confuses the router, since it assumes that the address of the system on the other end of the serial line keeps changing.

When DECnet routing is enabled, the system address changes from the original media access control (MAC) address to the DECnet-derived address, which is based on its area and node numbers. This address change affects the static routes on other bridge/routers configured to use this bridge/router as the next hop.



CAUTION: *If you enable DECnet routing on a path where you have reassigned the MAC address using LAN Address Administration (LAA), you may affect the DECnet address for that path. For more information on LAN Address Administration and how it may affect DECnet addresses, see the Configuring LAN Address Administration chapter.*



IDEcnet can not coexist with IP's VRRP feature on the same bridge/router.

- 3 Select the desired type of routing by entering one of the following commands.

To enable both intra- and interarea (level 2) routing, enter:

```
SETDefault -DECnet NodeType = Area
```

To enable intra-area (level 1) routing only, enter:

```
SETDefault -DECnet NodeType = RoutingIV
```

- 4 If any node in the area selected has a higher number than 255, increase the MaxNodeNumber parameter using:

```
SETDefault -DECnet MaxNodeNumber = <value>
```

- 5 Verify the DECnet configuration by entering:

```
SHow -DECnet CONFfiguration
```

The router displays the DECnet configuration information. If the CONTrol parameter is not set to route, or if the address that you just configured is incorrect, repeat steps 1 and 2.

For detailed information on these parameters, see the DECnet Service Parameters chapter in *Reference for Enterprise OS Software*.

To complete the configuration for PPP links, see the Configuring Wide Area Networking Using PPP chapter.



If DECnet routing is enabled after Internetwork Packet Exchange (IPX) routing, 3Com recommends flushing the existing IPX routing tables of adjacent IPX routers.

Configuring for Wide Area Networks

Routing DECnet over Frame Relay, Asynchronous Transfer Mode (ATM), Asynchronous Transfer Mode data exchange interface (ATM DXI), and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to route DECnet over Frame Relay, ATM, ATM DXI, or X.25 in a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM, ATM DXI, or X.25 cloud. For information on the number of virtual ports supported per platform, see Table 11 in the Configuring Advanced Ports and Paths chapter.

Routing DECnet over Switched Multimegabit Data Service (SMDS) is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your DECnet router to perform routing over SMDS, see the Configuring Wide Area Networking Using SMDS chapter.

To configure DECnet routing over PPP, see the Configuring Wide Area Networking Using PPP chapter. For information on wide area networking using Integrated Services Digital Network (ISDN), see the Configuring Wide Area Networking Using ISDN chapter.

Verifying the Configuration

Before you use a router to interconnect networks, verify that the routers you configured are recognized by the network and are functional by following these steps:

- 1 Check the routing table by entering:

```
SHoW -DECnet AllRoutes
```

The routing table displays all DECnet areas and nodes to which a router has access. Check to make sure that the routers and end nodes you configured appear in the routing table.

- 2 Check the status of the ports previously configured on your router by entering:

```
SHoW -DECnet STATUS
```

The DECnet status table displays the status of the ports for this router. Ports configured with DECnet routing enabled should be in the RUNNING state, which indicates that the port is operational.

For a description of other status states, see the DECnet Service Parameters chapter in *Reference for Enterprise OS Software*.

- 3 Check the values of the path parameters by entering:

```
SHoW -PATH CONFIguration
```

- 4 Check the current DECnet routing parameters using:

```
SHoW !<port> -DECnet CONFIguration
```

If the problem persists after these steps are taken, contact your network supplier or 3Com for assistance.

Getting Statistics

To view statistics, enter:

```
SHoW -SYS STATistics -DECnet
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information on these parameters, see the

Configuring DECnet Routing chapter in *Reference for Enterprise OS Software*. For information on interpreting the statistics displays, see the Statistics Displays appendix.

Troubleshooting the Configuration

If you are unable to make connections to nodes within the local area or nodes in other areas after setting up the router, review the following troubleshooting procedure. Using this procedure can correct problems in making single-hop (involving one router) and multiple-hop (involving more than one router) connections. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier for assistance.

To troubleshoot the DECnet configuration, follow these steps:

- 1 Check that all cables on all routers in a specific path in the routing table are properly connected and that the routers are properly installed.

For installation instructions, see the installation guide for your bridge/router.

- 2 Check that the state of the port is Up by entering:

```
SHow -PORT CONFIguration
```

If the state of the port is not Up, check that you have correctly completed the basic installation described in *New Installation for NETBuilder II Software*.

- 3 Check the status of the CONTROL parameter by entering:

```
SHow -DECnet CONTrol
```

The router displays the current values for the CONTROL parameter. If the CONTROL parameter for a port is set to NoROute, enable the DECnet router using:

```
SETDefault !<port> -DECnet CONTrol = ROute
```



Enable bridging and all routing protocols before enabling DECnet routing.

- 4 Check the status of the ports on your router by entering:

```
SHow -DECnet STATUS
```

The DECnet status table displays the status of the ports for this router. Ports configured with DECnet routing enabled should be in the RUNNING state, which indicates that the port is operational. If a port is in the DOWN state:

- Check the port and the associated path configuration to see if they are enabled.
- Enable the port and/or path if necessary.
- Check the cables along the associated path to ensure that they are properly connected.

For a description of other status states, see the DECnet Service Parameters chapter in *Reference for Enterprise OS Software*.

- 5 Check whether the node you are trying to reach is in the DECnet routing table by entering:

```
SHow -DECnet AllRoutes
```

The DECnet router displays the DECnet routing table entries. From the table entry, you can determine the path being used. Examine the entries to make sure a route in the table is taking the appropriate path.

Customizing the Configuration

This section provides additional procedures you can use to configure your DECnet router. For details on parameters, see the DECnet Service Parameters chapter in *Reference for Enterprise OS Software*.

Controlling Routing Information

The DECnet route filters allow you to control the routing information that the router advertises to or accepts from adjacent routers on a specified port. You can also control the list of adjacent routers on a specific port to send to or listen for routing information.

For a brief explanation of the route filtering parameters, see “Related Information” earlier in this chapter. For more information about the DECnet route filtering parameters, see the DECnet Service Parameters chapter in *Reference for Enterprise OS Software*.

To add a DECnet address, a list of addresses, or a range of addresses to the route list in the procedure that follows, use the ADD command. To exclude specific routes, use the ADD command with the tilde (~) prefix before each DECnet address to be excluded. For details on specifying lists and ranges of DECnet addresses, see the DECnet Service Parameters chapter in *Reference for Enterprise OS Software*.

Procedure

To define route filtering, follow these steps:



All the parameters in this procedure are port-specific.

- 1 Specify the routes advertised in routing updates using:

```
ADD !<port> -DECnet AdvertisePolicy <DECnet address>
```

To exclude a specific address, use:

```
ADD !<port> -DECnet AdvertisePolicy ~<DECnet address>
```

- 2 Specify the routes that are accepted from the routing updates of an adjacent router using:

```
ADD !<port> -DECnet ReceivePolicy <DECnet address>
```

- 3 Add a DECnet address to the list of adjacent routers that receive routing updates from this router using:

```
ADD !<port> -DECnet AdvToNeighbor <DECnet address>
```

- 4 Specify a list of trusted adjacent routers to listen for router hellos and routing updates using:

```
ADD !<port> -DECnet RcvFromNeighbor <comma-separated list of DECnet addresses>
```

- 5 Enable DECnet route filtering using:

```
SETDefault !<port> -DECnet PolicyControl = (AdvertisePolicy, ReceivePolicy)
```

Related Information

There are four route filtering parameters:

- AdvertisePolicy allows you to specify the routes that are advertised to adjacent routers in routing updates.

- ReceivePolicy allows you to specify the routes that are accepted from adjacent routers and cached in the routing tables.
- AdvToNeighbor allows you to specify the adjacent routers where routing updates may be sent.
- RcvFromNeighbor allows you to specify from which adjacent routers to accept hellos and routing updates.

When all four routing policies are configured and enabled, the following route filtering occurs:

- Before a routing update is transmitted onto the outbound port, the local routing information is filtered by the AdvertisePolicy parameter. This filtered information then is sent to the set of adjacent routers specified by the AdvToNeighbor parameter.
- When a router receives a routing update, only routing updates reported by the set of adjacent routers specified by the RcvFromNeighbor parameter are accepted. These routing updates then are filtered by the ReceivePolicy parameter before the reported routes are cached in the local routing database.

To enable and disable route filtering, use the PolicyControl parameter.

Setting the Priority

The PRIOrity parameter changes the priority of the router on the LAN. The router with the highest priority is elected as the designated router on the attached LAN. If multiple routers on the LAN have the highest priority, the router with the highest node ID is elected as the designated router.

To set the router priority, use:

```
SETDefault !<port> -DECnet PRIOrity = <number> (1-127)
```

Setting the Cost

The COST parameter allows you to change the route cost associated with the attached network. For DECnet Phase IV routing, packets are forwarded to the destination using the least-cost route.

To specify the cost associated with a network, use:

```
SETDefault !<port> -DECnet COST = <number> (1-25)
```

Enabling and Disabling Triggered Routing Updates

The CONTrol parameter allows you to choose triggered or complete routing updates as well as enabling and disabling routing.

Triggered updates occur whenever the routing table changes. Complete routing updates occur at intervals determined by the setting of the RoutingTime parameter (see "Setting the Routing Time"). Complete updates are always sent at regular intervals, regardless of the Trigger/NoTrigger setting.

To select triggered routing updates, use:

```
SETDefault !<port> -DECnet CONTrol = Trigger
```

To deselect triggered routing updates, use:

```
SETDefault !<port> -DECnet CONTrol = NoTrigger
```

Setting the Routing Time

The RoutingTime parameter allows you to specify the timer interval (in seconds) at which the router sends complete routing updates to adjacent router nodes.

To set the routing time, use:

```
SETDefault !<port> -DECnet RoutingTime = <seconds>(5-65535)
```

Setting the Hello Messages Time

The HelloTime parameter sets the frequency at which the router sends hello messages to adjacent nodes. The value of the HelloTime parameter also determines the value of time-to-live (TTL) as seen by its adjacent nodes. The TTL of an adjacent node is based on its HelloTime parameter value. The formula used to calculate time-to-live is given in "Related Information."

Procedure

To set the HelloTime parameter, use:

```
SETDefault !<port> -DECnet HelloTime = <seconds>(5-8191)
```

Related Information

The following formula is used to calculate TTL:

$$\text{TTL} = K * \text{<value of HelloTime parameter>}$$

where:

K = 2 if the adjacent node is on a serial line

or

K = 3 if the adjacent node is on a LAN

For example, if the value of the HelloTime parameter configured for the adjacent node on a LAN is 30 seconds, then the TTL for the adjacent router is 90 seconds. If the local node does not receive a hello message from the adjacent node before the TTL counts down from 90 to 0 seconds, the adjacent node is declared down.

How the DECnet Router Works

This section provides information on how the DECnet router works.

DECnet Network

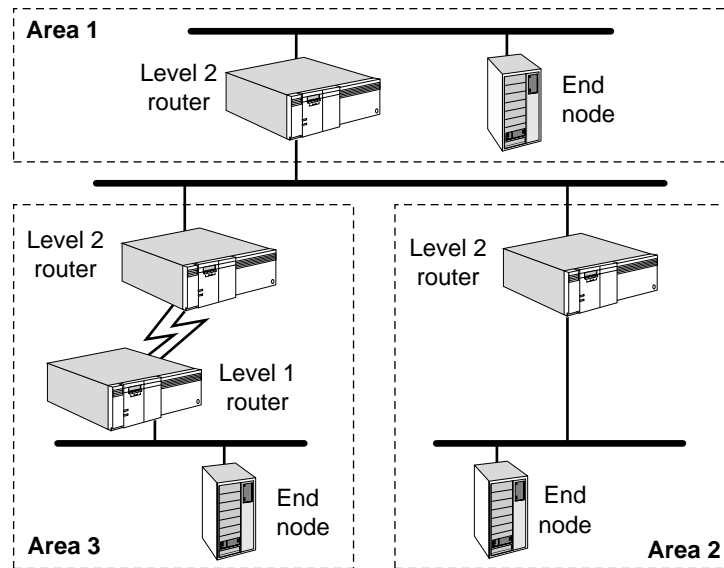
A DECnet network is configured on each port where DECnet packets are received and sent. The port can be any LAN or WAN port.

A DECnet network consist of nodes that do and do not route packets. Nodes that do not route packets (a host such as a VAX station, for example) are called end nodes. Nodes that route packets are called routers. One router per LAN (Ethernet, token ring, Fiber Distributed Data Interface (FDDI)) has the additional role of routing packets for the end nodes on that LAN. These nodes are called designated routers. The PRIOrity parameter can be used to force a particular router to be the designated router on the LAN. For more information on this parameter, see "Customizing the Configuration" earlier in this chapter.

Routing nodes must be configured (with the NodeType parameter) as either Level 1 or Level 2 routers. A Level 1 router routes packets within a local area only. A Level 2 router routes packets both within a local area (intra-area) and between areas (inter-area).

Figure 212 is an example of a DECnet network. This DECnet network is composed of four LANs, which are separated into three areas.

Figure 212 DECnet Network



A router must check its routing table to determine where to route a packet.

- If the destination node is within a local area, the router can forward the packet directly to the destination node or to the next-hop router, if appropriate, which sends it to the destination node.
- If the destination node is in another area, the router sends the packet to the nearest Level 2 router in the local area, which sends it to a Level 2 router in the other area. The Level 2 router in the other area then forwards the packet to the destination node in the same way that a Level 1 router does.

Routing Tables Display the DECnet routing table using the AllRoutes parameter.

A Level 2 router displays two types of routing tables: the DECnet Level 1 (intra-area) routing table and the DECnet Level 2 (interarea) routing table. The DECnet Level 1 routing table displays information on nodes located within the local area that the router can reach. The DECnet Level 2 routing table displays information on other areas that the router can reach. A Level 1 router displays the DECnet Level 1 routing table only.

Each entry in the DECnet Level 1 routing table includes the following types of information that determine how a packet is routed:

- Reachable intra-area destination (node and port)

The DECnet node address and port number of reachable intra-area destinations.
- Next hop

The DECnet address of the next router to which a packet is forwarded on its way toward its destination.
- Cost

The cost value associated with using the indicated intra-area route. In a DECnet network, packets are routed to their destination using the route with the smallest total cost. The COST parameter configures the cost value for each

port. The route cost indicates the total cost of traversing one or more network interfaces to reach the intra-area destination.

- Number of hops between router and destination

The number of hops is equal to the number of routers traversed to reach the destination node.

- BlkSize

The maximum packet size that can be sent to that end node.

- Priority

The priority of the router on the LAN. The priority determines which router on the LAN will be the designated router. The designated router is the router with the highest priority. If two or more routers have the highest priority, the router with the highest node ID becomes the designated router.

- TTL

Indicates the time-to-live in seconds before the route is removed from the routing table. The HelloTime parameter configuration of the adjacent router or end node controls the TTL. For details on this parameter, see "Customizing the Configuration" earlier in this chapter.

Each entry in the DECnet Level 2 routing table includes the following types of information, which determine how a packet is routed:

- Reachable interarea destination (area and port)

The DECnet area number and port number of reachable interarea destinations.

- Next hop

The DECnet address of the next router to which a packet is forwarded for routing to its area destination.

- Cost

The cost value associated with using the indicated interarea route. In a DECnet network, packets are routed to their destination using the route with the smallest total cost. The COST parameter configures the cost value for each port. The route cost indicates the total cost of traversing one or more network interfaces to reach the interarea destination.

- Number of hops between router and destination

The number of hops is equal to the number of area routers traversed to reach the destination node.

- TTL

Indicates the time-to-live in seconds before the area route is removed from the routing table. The value of the adjacent router's HelloTime parameter controls the TTL. For details on this parameter, see "Customizing the Configuration" earlier in this chapter.

The DECnet Level 2 routing table also summarizes the number of reachable areas, nodes within the local area, adjacent routers, and adjacent end nodes.

When the router learns multiple routes for a node or area, the least-cost route is always used to reach the node or area. For information on how the router makes the routing decision, see "Cost-effective Routing" earlier in this chapter.

Learning Routes

A router learns routes through routing update messages. These messages update the routing tables with all known destinations and their associated costs and numbers of hops.

Routing update messages are propagated throughout the network in the following manner:

- A node sends a routing update to an adjacent node (a node that is one logical hop away).
- When this adjacent node receives the routing update, it compares the information in the routing update with the information in its routing table.
- If the information in the routing update results in route changes in the routing table and the triggered update option is selected, a routing update with the new route information is generated and sent to the adjacent routers.
- Routing information changes are propagated to all router nodes on the network in this manner.

Level 1 routers send and receive messages to and from all adjacent nodes within the same area. Level 2 routers send and receive messages to and from all adjacent nodes within the same area as well as to and from adjacent Level 2 routers in other areas.

Complete routing updates are sent at user-configured time intervals. The frequency at which routing updates are sent is configured with the `RoutingTime` parameter. For more information on this parameter, see “Customizing the Configuration” earlier in this chapter.

However, if you have selected triggered routing updates and a router detects a change in the topology of your network (for example, a node is not operating), a routing update immediately reports to the adjacent routers that this node is unreachable. See “Enabling and Disabling Triggered Routing Updates” earlier in this chapter.

Network Reachability and Split Horizon

A node is considered *reachable* when the computed cost and number of hops it takes to reach is less than the maximum cost and the maximum number of hops you configured for a router. To determine which nodes are reachable, check the routing table for each router.

The values that you set for the `MaxCost`, `MaxHops`, `MaxAreaCost`, and `MaxAreaHops` parameters determine the maximum cost and number of hops allowed for a node before the node is deemed unreachable.

The DECnet router avoids routing loops using *split horizon*. Split horizon prevents routing loops that may occur when a node includes information on other nodes learned from the same interface on which the routing update is sent. A DECnet router automatically uses split horizon with poison reverse by marking a route as unreachable in a routing update sent on the same interface from which the route was learned. Split horizon occurs automatically and requires no configuration.

Figure 213 illustrates how split horizon is used in DECnet routing. In this configuration, router A sends a routing update on port 1 that includes the following information:

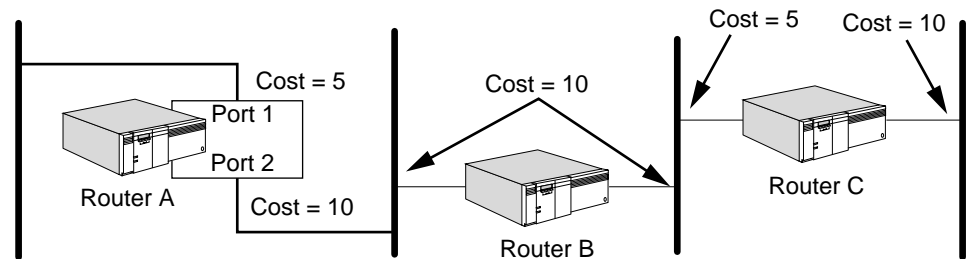
- Router A is 0 hops away and has a cost of 0 (since this is information it is reporting on itself).
- Router B is 1 hop away and has a cost of 10.
- Router C is 2 hops away and has a cost of 20.

The routing message that router A sends on port 2 includes the following information:

- Router A is 0 hops away and has a cost of 0 (since this is information it is reporting on itself).
- Router B is 31 hops away and has a cost of 1023 (unreachable).
- Router C is 31 hops away and has a cost of 1023 (unreachable).

Split horizon prevents a router from advertising networks to any router it learned of those networks from. In this example, router A does not advertise to router B the route to router C. If the connection from router B to router C fails, split horizon prevents router B from sending packets bound for router C to router A.

Figure 213 DECnet Routing Using Split Horizon



Cost-effective Routing

The DECnet router supports cost-effective routing, which means that the router selects the route with the lowest cost. The lowest-cost route is not necessarily the shortest (the route with the fewest hops). For example, imagine that two routes to another area exist. Route A requires three hops and has an associated cost of 30. Route B requires four hops and has an associated cost of 25. Route B would be selected because it incurs the least cost (it is the most cost-effective route), although it requires more hops.

If a router has two routes with the same cost associated with each route, the router forwards packets to the router with the higher node ID.

Only the most cost-effective route appears in the routing tables.

Routing Phase IV Traffic over DOD Lines

For DECnet Phase IV environments where traffic is routed over dial-on-demand lines, routing updates and periodic hellos are suppressed once the router adjacency is established, and the routing database is synchronized. This allows the demand circuit to be brought down when it is not carrying traffic.

Address Translation Gateway Support

The Address Translation Gateway feature provides internetwork routing support and address translation for DECnet networks.

Internetwork Routing Support

A DECnet router can support one or more independent DECnet Phase IV networks attached to its LAN or WAN interfaces. Connectivity between the attached DECnet Phase IV networks is achieved through address translation.

Address Translation

Address translation allows connectivity between specific DECnet nodes on different networks that otherwise cannot communicate because of address conflicts between networks. Defining address translation for specific nodes on the

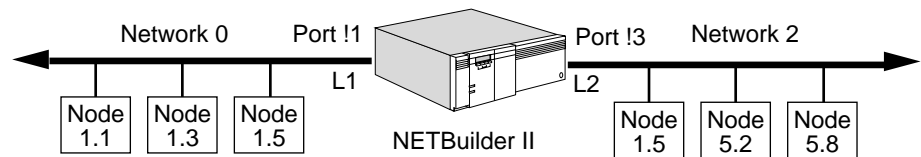
selected networks resolves duplicate addresses and permits internetwork communication.

Address Translation Configuration Example

In the following sample configuration, a NETBuilder II bridge/router is attached to DECnet Phase IV networks.

The router is connected to network 0 through Ethernet port 1 as a Level 1 intra-area router with an address of 1.2. The router is also connected to network 2 through Ethernet port 3 as a Level 2 area router with an address of 5.1. Both networks are independent until the internetwork routing function is enabled through the user-defined address translation map.

Figure 214 DECnet Address Translation Configuration



The following diagram illustrates the address conversion in a packet exchange between node 1.5 on network 0 and node 5.8 on network 2.

```

SA=1.5 / DA=1.8
----->
                                     SA=3.1 / DA=5.8
                                     ----->
                                     SA=5.8 / DA =3.1
                                     <-----
SA=1.8 / DA=1.5
<-----

```

The following diagram illustrates the address conversion in a packet exchange between node 1.1 on network 0 and node 1.5 on network 2.

```

SA=1.1 / DA=1.9
----->
                                     SA=3.2 / DA=1.5
                                     ----->
                                     SA=1.5 / DA =3.2
                                     <-----
SA=1.9 / DA=1.1
<-----

```

Without the above address map, node 1.1 on network 0 cannot communicate with node 1.5 on network 2, because of address conflicts between the networks.

To configure the sample address translation configuration, follow these steps:

- 1 Configure DECnet routing for network 0 by entering:

```

SETDefault -DECnet ADDRESS = 1.2
SETDefault -DECnet NodeType = RoutingIV
SETDefault -DECnet MaxNodeNumber = 512
SETDefault !1 -DECnet CONTROL = ROute

```

- 2 Configure DECnet routing for network 2 by entering:

```

SETDefault -DECnet ADDRESS = 5.1 2

```



```

SETDefault -DECnet NodeType = Area 2
SETDefault -DECnet MaxAreaNumber = 7 2
SETDefault !3 -DECnet Network = 2
SETDefault !3 -DECnet Control = ROute

```

- 3 Configure address translations between network 0 and network 2 by entering the following commands.

Map virtual node 1.9 on network 0 to real node 1.5 on network 2 by entering:

```
ADD -DECnet AddressMap 1.9@0 1.5@2
```

Map virtual node 3.2 on network 2 to real node 1.1 on network 0 by entering:

```
ADD -DECnet AddressMap 3.2@2 1.1@0
```

Map virtual node 1.8 on network 0 to real node 5.8 on network 2 by entering:

```
ADD -DECnet AddressMap 1.8@0 5.8@2
```

Map virtual node 3.1 on network 2 to real node 1.5 on network 0 by entering:

```
ADD -DECnet AddressMap 3.1@2 1.5@0
```

The above address translation map allows nodes 1.1 and 1.5 on network 0 to communicate with nodes 1.5 and 5.8 on network 2.

- 4 To enable the configured address map to allow internetwork routing, enter:

```
SETDefault -DECnet InterNetRoute = AddressMap
```

Because the router does Level 1 intra-area routing on network 0, node 1.5 on network 0 cannot communicate with node 5.8 on network 2 without an address map. By defining node 5.8 as the virtual node 1.8 on network 0, node 1.5 can access node 5.8 by connecting to the virtual node 1.8 on network 0.

A packet received from network 0 and destined for the virtual address 1.8 will result in the conversion of the real address 5.8. The next hop to 5.8 is determined by a lookup in the routing table for network 2. The source address is translated to its virtual address on network 2 and the packet is forwarded.

Virtual addresses 1.8 and 1.9 are advertised to network 0 as reachable nodes with zero cost/hop. The virtual area 3 is also advertised to network 2 as a reachable area with zero cost/hop.



The user-defined virtual address must not already exist in the associated network.

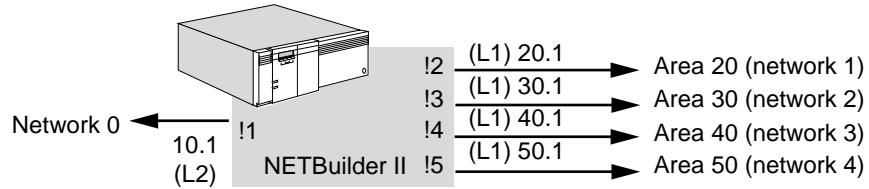
Because only configured virtual addresses are advertised to their associated network, only nodes that exist in the address translation map on both networks can communicate directly. In the sample configuration, node 1.3 on network 0 cannot communicate with any nodes on network 2. Node 5.2 also cannot access any nodes on network 0.

Internetwork Boundary Routing

Internetwork Boundary Routing software architecture allows connectivity between DECnet nodes in a Boundary Routing environment where each of the remote networks resides in a different DECnet area.

In this sample configuration (Figure 215), the central router is connected to network 0 on Ethernet port 1 as an area router with the address 10.1. The router is also connected to remote networks 1 through 4 through PPP links.

Figure 215 DECnet Internetwork Boundary Routing Configuration



Each of these networks exists as an independent network until the internetwork boundary is enabled.

To enable internetwork Boundary Routing, enter:

```
SETDefault -DECnet InterNetRoute = 0, 1, 2, 3, 4
```

All nodes on networks 0, 1, 2, 3, and 4 can now connect to each other. The router advertises areas 20, 30, 40 and 50 as reachable areas to network 0. Packets received from the remote networks that are destined to nodes on one of the networks configured for internetwork Boundary Routing are forwarded to that network if the destination is reachable.

Phase IV to Phase V Transition Support

The DECnet Phase V gateway provides coexistence and interoperability of DECnet Phase IV and Phase V (Open Systems Interconnect) nodes in a DECnet network. For more information, see "DECnet Phase V and Phase IV Terms" earlier in this chapter.

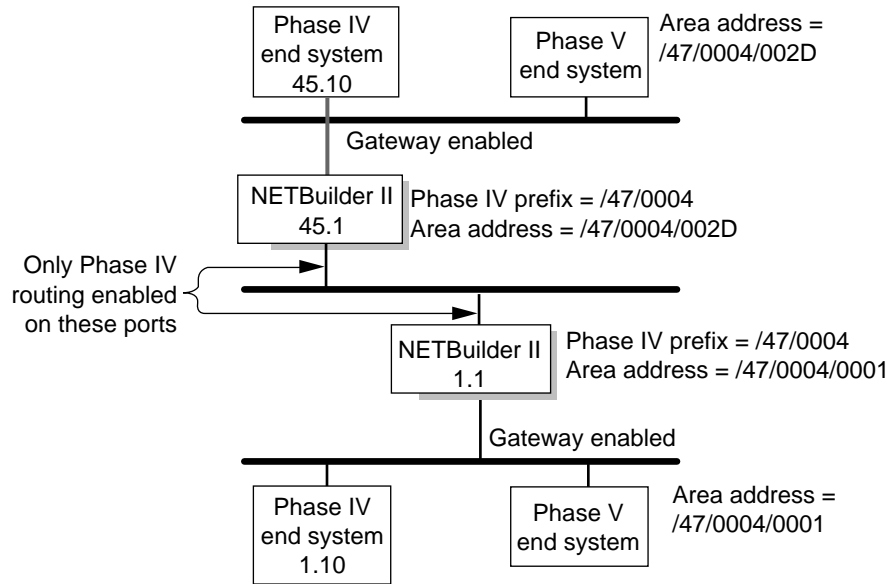
The following features permit interoperability: Phase IV to Phase V Translation and DECnet area to pseudo areas translation.

Phase IV to Phase V Translation

The DEC-compatible Phase IV to Phase V translation algorithm on addressing, data packet, and route advertisements is supported by the 3Com Phase IV to Phase V Transition Support feature. The translation allows Phase IV hosts to exist in Phase V networks and Phase V hosts to exist in Phase IV networks. The Phase IV hosts can communicate only with Phase V hosts that have Phase IV-compatible addresses. A Phase IV-compatible address is a Phase V address that is within the Phase IV addressing limits.

In Figure 216, Phase IV and Phase V end systems can communicate with each other using Phase IV routing, Phase V routing, or a combination of Phase IV and Phase V routing. The gateway provides the common routing path that enables these end systems to communicate in the same area or in different areas.

Figure 216 DECnet Phase IV to Phase V Translation

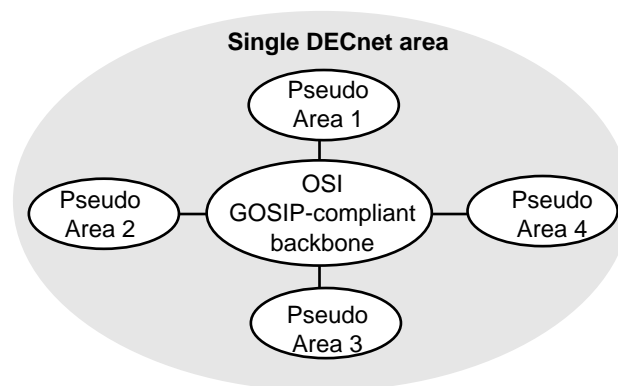


The 3Com implementation supports Phase IV and Phase V routing protocols in a single DECnet area. Supporting both protocols in a single area allows Phase V support to be added to a Phase IV area without modifying the existing Phase IV support. The 3Com router translates the routing information between the Phase IV and the Phase V routing environments. In Phase IV routing updates, 3Com routers advertise reachability to Phase V hosts that have a Phase IV-compatible address. 3Com routers also advertise reachable Phase IV hosts in Phase V Link State advertisements.

DECnet Area to Pseudo Areas Translation

A DECnet Phase V area that is Phase IV-compatible can be subdivided into multiple pseudo areas with a smaller address space. The pseudo areas allow a unique OSI area address to be assigned to each DECnet site within the common DECnet area (see Figure 217). This permits intersite communication through Level 2 routing (with static prefix routes) in a GOSIP-compliant OSI backbone network.

Figure 217 DECnet Pseudo Areas



When multiple DECnet sites share a single DECnet area, and connectivity between Phase IV and Phase V hosts must be maintained, to add Phase V routing support

you need to configure all sites to reside in the same OSI area with the following area address:

```
<common Phase IV NSAP Prefix/common DECnet area ID>.
```

Sites that are connected to a GOSIP-compliant OSI backbone network require routing domain boundaries to restrict routing information exchanges. The result is the partitioning of the common OSI area into disjoint subareas.

Because Phase V nodes currently supports multihoming to only three area addresses, a loss of connectivity in the partitioned area may result. In this case, a pseudo area can be assigned to each site to work around the routing problem in the partitioned OSI area. The pseudo area address of a site, formed by concatenating the common pseudo area prefix and the pseudo area ID of the site, is unique in the common OSI area. This pseudo area address allows intra-area traffic of a site destined for another site to be routed across the backbone's routing domain boundary to the destination site using the backbone's Level 2 interarea routing.

When a packet is forwarded to a remote site, at each site the router maps the destination network service access point (NSAP) address that is within the address space of the common OSI area into its corresponding pseudo area address for intersite routing. When a packet is received from the OSI backbone, a destination pseudo area address is converted into its corresponding NSAP address for intrasite routing. The pseudo area addresses are used strictly for routing intra-area traffic across a partitioned area. The remote pseudo area addresses must be configured on a 3Com router as reachable NSAP address prefixes, using the PrefixRoute parameter in the ISIS Service. For more information about NSAP addressing, see the NSAP and PSAP Addressing appendix.

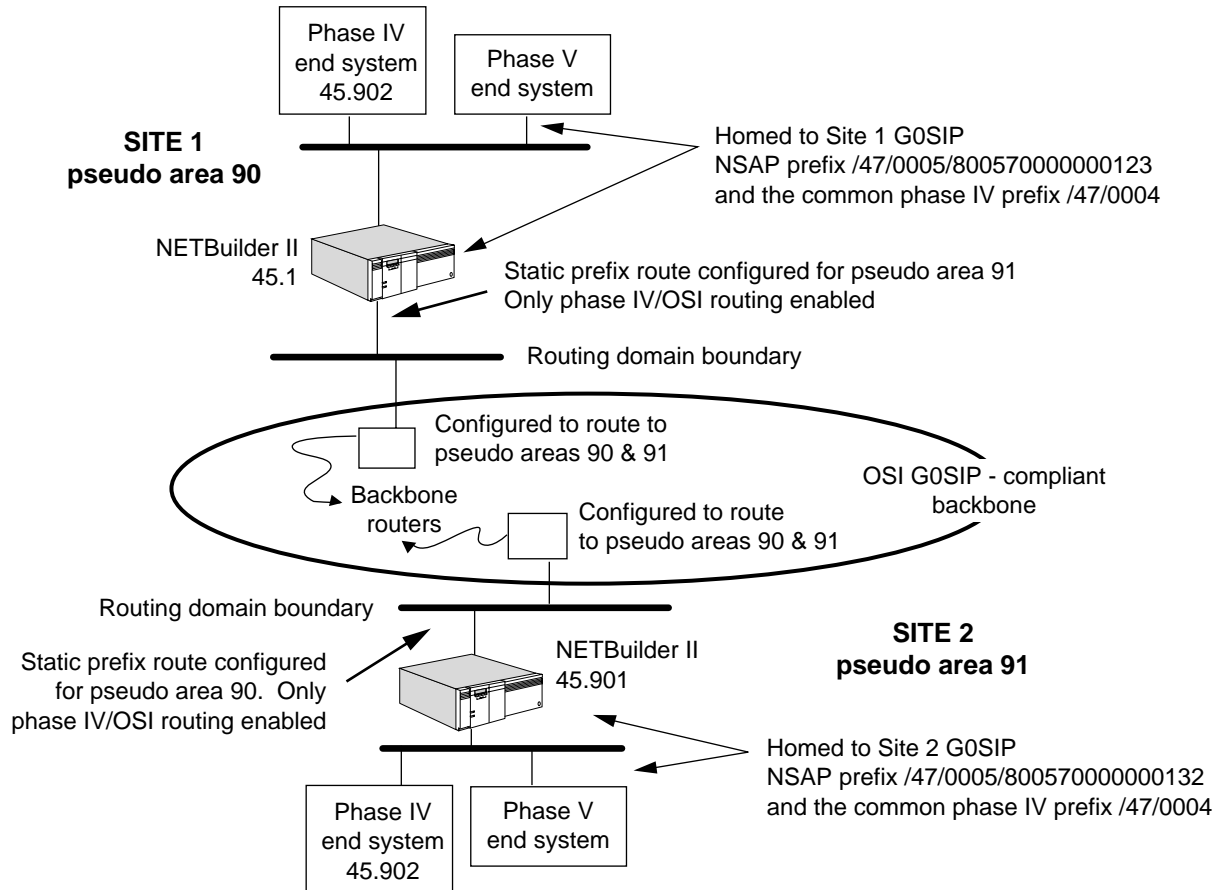
Pseudo Area Configuration

In Figure 218, sites 1 and 2 share the same DECnet area 45. Both sites are configured to support two pseudo areas. Site 1 is configured in pseudo area 90. In this area, node addresses 45.1 through 45.511 are mapped to addresses 90.1 through 90.511. The Phase IV end systems and Phase V end systems with a Phase IV-compatible address at both sites can communicate through the OSI GOSIP-compliant backbone.

Site 2 is configured in pseudo area 91 in which node addresses 45.512 through 45.1023 are mapped to addresses 91.1 through 91.511. The Phase IV and Phase V

end systems with a Phase IV-compatible address at both sites can communicate through the OSI GOSIP backbone.

Figure 218 DECnet Pseudo Area Configuration



Phase IV to Phase V Transition Configuration Example

To configure Phase IV to Phase V transition on NETBuilder 45.1 based on the example in Figure 218, follow these steps:

- 1 Configure DECnet Phase IV routing by specifying the DECnet address to be used by the router. Enter:

```
SETDefault -DECnet ADDRESS = 45.1
```

- 2 Specify the node type. In the following command, the node type is Area. Enter:

```
SETDefault -DECnet NodeType = Area
```

- 3 Enable DECnet routing on ports 1 and 2 by entering:

```
SETDefault !1 -DECnet CONTROL = ROute
SETDefault !2 -DECnet CONTROL = ROute
```

- 4 Configure Phase V OSI routing.

The area ID field of the local OSI area address must match the local DECnet area number.

When the DECnet gateway function is enabled, the area address, formed by concatenating the IVPrefix and the area number of the local DECnet Phase IV

address, must match one of the area addresses configured for the OSI router. Enter the following commands to set the OSI intermediate system area address.

- a To specify the area address, enter:

```
ADD -ISIS AreaAddress /47/0004/002D
```

The NSAP address is specified in hexadecimal format. The area id %002D in the NSAP address matches the local DECnet decimal area number 45.

- b To specify the intermediate system as a Level 2, enter:

```
SETDefault -ISIS Mode = Level2
```

- c To enable the CLNP routing function, enter:

```
SETDefault -CLNP CONTROL = Route
```

The DECnet Phase IV address is specified in decimal format while the OSI area address is specified in hexadecimal format.

- 5 Configure Phase IV to Phase V translation.

The Phase IV NSAP prefix must match the area prefix of an existing OSI area address configured for the OSI router.

- a To specify the common Phase IV NSAP Prefix, enter:

```
SETDefault -DECnet IVPrefix = /47/0004
```

- b To enable the DECnet Phase IV to Phase V translation, enter:

```
SETDefault -DECnet GatewayControl = GateWay
```

- 6 Configure the pseudo area mapping.

In this example, the local pseudo area is 90 and the remote pseudo area is 91.

```
SETDefault -DECnet PseudoAreaPrefix = /47/0005/8000570000000123
```

```
SETDefault -DECnet MaxPseudoAreas = 2
```

The following information is displayed:

```
Local Pseudo Area address: /47/0005/80005700000001230090
```

```
Route Pseudo Area address: /47/0005/80005700000001230091
```



The MaxPseudoAreas configuration must be identical for all communicating pseudo areas.

- 7 Configure a static prefix route for the remote pseudo area 91 by entering:

```
SETDefault !1 -ISIS PrefixRoute /47/0005/80005700000001230091
```

```
%080001020304
```

- 8 Enable the pseudo area translation by entering:

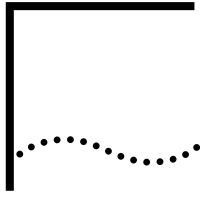
```
SETDefault -DECnet GatewayControl = PseudoArea
```

DECnet Phase V and Phase IV Terms

This section describes DECnet-specific terms:

| | |
|----------------|--|
| DECnet Phase V | OSI-compatible. Phase V routing conforms to the ISO's CLNP, ES-IS, and IS-IS protocols. In addition, Phase V nodes are backward-compatible with Phase IV nodes. A Phase V node determines the packet format to use with an adjacent node based on the type of hello message received from that node. |
|----------------|--|

| | |
|----------------------------------|--|
| Phase IV NSAP Prefix | <p>The common NSAP Prefix. This prefix must be used in Phase V routing environments to allow communication between Phase IV and Phase V systems in a routing domain. A 3Com router serving as a DECnet gateway concatenates the configured Phase IV NSAP Prefix and its own Phase IV ArealD to form the Phase IV OSI area address for advertising reachable Phase IV nodes in Phase V areas.</p> |
| Phase IV-compatible NSAP Address | <p>DECnet Phase V nodes can communicate with DECnet Phase IV nodes through a 3Com router serving as a DECnet gateway when the Phase V node is configured with a Phase IV-compatible NSAP address. A Phase IV-compatible NSAP address is defined as:</p> <p><NsapPrefix/ArealD/StationID/Selector>.</p> <p>A Phase IV-compatible NSAP address assures that the address can be translated from Phase V to Phase IV and back again without change. A Phase IV-compatible NSAP address must conform to the following rules defined by DEC:</p> <ul style="list-style-type: none">■ The NSAP Prefix must match the Phase IV NSAP Prefix specified for the 3Com router with the IVPrefix parameter.■ The 2-octet ArealD has a value within the range of 0 to 63 and matches the low order 6 bits of the 6-octet StationID.■ The high order 32 bits of the StationID must match the DECnet architectural constant <i>AA-00-04-00</i> (hexadecimal). |
| Phase IV-compatible Area Address | <p>Reachability information of Phase V nodes that are configured with a Phase IV-compatible NSAP address is advertised by the 3Com router to adjacent Phase IV routers. A reachable Phase IV-compatible area address, <NsapPrefix/ArealD>, is advertised by 3Com routers as a reachable Phase IV area if:</p> <p>The NSAP Prefix matches the Phase IV NSAP Prefix specified for the 3Com router with the IVPrefix parameter.</p> <p>The 2-octet ArealD has a value within 0 to 63.</p> |



CONFIGURING OSI ROUTING

This chapter describes how to configure, customize, and troubleshoot Open Systems Interconnection (OSI) routers.



For conceptual information, see “How the OSI Router Works” later in this chapter.

Setting Up a Basic OSI Router

The procedures in this section describe the minimum steps required to enable your system to route OSI packets. Depending on your network requirements, you may want to further configure the router according to later sections in this chapter.

Configuring for Local Area Networks and Point-to-Point Protocol Links

Use this procedure to configure basic OSI routing for LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter, and log on to the system with Network Manager privilege.
- It is assumed that you are familiar with the protocols supported by the router. See ISO 8473 for information on connectionless mode network service, ISO 9542 for information on the End System-to-Intermediate System (ES-IS) Protocol, ISO 10589 for information on the Intermediate System-to-Intermediate System (IS-IS) routing Protocol, and ISO 8348, Addendum 2, for NSAP addressing.
- If you are using DECnet routing with OSI routing, you must configure DECnet routing before OSI routing. Configuring DECnet routing changes the Ethernet address of the router, and OSI routing protocols will not recognize the new Ethernet address.
- To configure the OSI router, you must set some parameters in the ISIS and CLNP Services.
- OSI cannot coexist with IP's VRRP feature on the same bridge/router.

Procedures

To configure the bridge/router to perform basic OSI routing, follow these steps:

- 1 Determine the area address of the router.

3Com bridge/routers are shipped with the default area address of /49/0053.

 - a To display the current area address of the router, enter:
SHoWDefault -ISIS AreaAddress
 - b To configure the area address for the router, use:

```
ADD -ISIS AreaAddress <NSAP address>
```

For example, if you want to reset the area address of a router to /47/0004/00351100, enter:

```
ADD -ISIS AreaAddress /47/0004/00351100
```

Guidelines exist for setting area addresses. For more information on this topic, see “Area Addresses” later in this chapter.

- c After changing or adding additional area addresses, delete any old area addresses.

For example, to delete the default area address, enter:

```
DELEte -ISIS AreaAddress /49/0053
```

You can configure up to three area addresses. Multiple area addresses are normally used when transitioning your network from one configuration to another. For example, multiple area addresses can be used if you are introducing a new area address to replace an old one, you are merging two areas into one, or you are separating one area into two areas.

- 2 Determine whether a router is to perform as a Level 2 router; if necessary, configure it to perform as a Level 2 router.

The default routing type is Level 1 (routing within an area or intra-area routing) only.

To configure the router to perform Level 2 routing (routing between areas or interarea routing), enter:

```
SETDefault -ISIS MODE = Level2
```

A router that is configured as a Level 2 router performs both intra-area and interarea routing.

- 3 Determine which ports are to be used for ISIS routing.

ISIS routing is enabled by default on all ports. To disable ISIS routing on a particular port on which you do not want ISIS routing to occur, use:

```
SETDefault !<port> -ISIS CONTrol = Disable
```

- 4 Enable the Connectionless Network Protocol (CLNP) routing function by entering:

```
SETDefault -CLNP CONTrol = Route
```

Enabling the routing function immediately starts the operations of both ES-IS and IS-IS routing protocols. The router becomes an IS, and it starts sending intermediate system hello (ISH) packets to the attached networks. Conversely, if the routing function is disabled, operations of both ES-IS and IS-IS routing protocols immediately stop.

- 5 If you have end systems that do not support the ES-IS Protocol, and the router needs to route packets to them, configure static routes for them by using:

```
ADD !<port> -CLNP ES <NSAP address> <SNPA>
```

For example, the following command adds a subnetwork point of attachment (SNPA) end system address for port 3:

```
ADD !3 -CLNP ES /47/0004/0035110008000200369101 %080002A01459
```

The port referenced in the command is the one where the end system (ES) is reachable. <NSAPaddress> is the NSAP address of the destination ES; <SNPA> is

the MAC address of the ES, or may be the MAC address of another router through which the ES is reachable.

6 Configure an interdomain route using:

```
ADD !<port> -ISIS PrefixRoute
```



This step applies to Level 2 routers at a routing domain boundary only.

For more information on how to set up interdomain routing, see “Setting Up Interdomain Routing” later in this chapter.

To complete the configuration for PPP links, see the Configuring Wide Area Networking Using PPP chapter.

Configuring for Wide Area Networks

You can configure the OSI router to perform routing over wide area network ports using Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), Switched Multimegabit Data Service (SMDS), X.25, and Integrated Services Digital Network (ISDN). To configure your OSI router to perform routing over Frame Relay, ATM DXI, SMDS, or X.25, see the Configuring Wide Area Networking Using Frame Relay chapter, the Configuring Wide Area Networking Using the ATM DXI chapter, the Configuring Wide Area Networking Using SMDS chapter, or the Configuring Wide Area Networking Using X.25 chapter. For information on wide area networking using ISDN, see the Configuring Wide Area Networking Using ISDN chapter.

Verifying the Configuration

This section describes how to verify the router configuration, check with OPING, examine statistics, and check its overall status. Before you use the router for interconnecting networks, check to see whether it can route packets properly. Send packets from one network to another to see if they are properly forwarded.

To verify the router configuration, follow these steps:

1 Check whether all the ESs on the directly attached networks are included in the End System Table by entering:

```
SHow -CLNP ES
```

Check the network attachment for any ESs that are not included in the table. For more information on this table, see “End System Table” later in this chapter.

2 Check whether all the intermediate systems (ISs) on the directly attached networks are included in the Intermediate System Table by entering:

```
SHow -CLNP IS
```

Check the network attachment for any ISs that are not included in the table. For more information on this table, see “Intermediate System Table” later in this chapter.

3 Check whether all ISs on the directly attached networks have established an adjacency with this router by entering:

```
SHow -ISIS ADJacencies
```

Compare the entries in the displayed adjacency table with the entries in the Intermediate System Table. All Level 2 ISs should be adjacent with each other, with adjacency type L2ONLY. All Level 1 ISs with area addresses in common should be adjacent with each other, with adjacency type L1ONLY. (Level 1 ISs with different

area addresses do not establish adjacencies with each other.) Neighboring routers configured with different Hello passwords are not adjacent.

4 Check the Level 1, Level 2, and Interdomain Routing Tables.

a To display the Level 1 Routing Table, enter:

```
SHoW -ISIS L1Route
```

The Level 1 Routing Table summarizes all reachable systems, both ESs and ISs, within the area. Check to make sure this table displays all ESs and ISs within the area.

b To display the Level 2 Routing Table, enter:

```
SHoW -ISIS L2Route
```

The Level 2 Routing Table applies only to routers that are configured as Level 2 routers. This table summarizes all reachable areas within the routing domain. Check to make sure that all areas are included.

c To display the Interdomain Routing Table, enter:

```
SHoW -ISIS PrefixRoute
```

The Interdomain Routing Table applies only to routers configured as Level 2. This table summarizes all reachable routing domains outside of this routing domain.

5 Examine the configuration of ports by entering:

```
SHoW -PORT CONFIguration
```

6 Examine the configuration of paths by entering:

```
SHoW -PATH CONFIguration
```

7 Examine the CLNP configuration and the ES and IS tables by entering:

```
SHoW -CLNP CONFIguration
```

8 Examine the ESIS configuration by entering:

```
SHoW -ESIS CONFIguration
```

9 Examine the ISIS configuration and adjacency table by entering:

```
SHoW -ISIS CONFIguration
```

10 Examine the Level 1 Routing Table by entering:

```
SHoW -ISIS L1Route
```

11 Examine the Level 2 Routing Table by entering:

```
SHoW -ISIS L2Route
```

12 Show a collective listing of all routing domains that can be reached from a particular routing domain by entering:

```
SHoW -ISIS PrefixRoute
```

Checking Packet-Forwarding Process

After you have configured your routers for OSI, check to see if they can forward packets properly.

To check to see if your routers are configured properly, follow these steps:

- 1 Select one router in your network and attach a terminal to its console port.
- 2 Use the OPING command to verify proper routing to each of the other routers:

For example, to send an echo request message to a router having the NSAP address /47/0004/0035130008000200182400, enter:

```
OPING /47/0004/0035130008000200182400
```

You may receive one of the messages in Table 52.

Table 52 OPING Command Messages

| Message | Meaning |
|---|---|
| Pinging ... destination is alive | Successfully reached destination: bidirection verified. |
| dest unreachable according to local routing table | The local router has no route. |
| Pinging... received Error Report PDU code 128 | The local router has either a default or a Level2 route, but the path to the destination is not complete. |

- If an error report protocol data unit (PDU) code is received, use the OTraceRoute command to determine where the route fails.

For example, to trace the path to the destination /47/0004/0035130008000200182400, enter:

```
OTraceRoute /47/0004/0035130008000200182400
```

You will receive this message:

```
TTL                Next_Hop_Address
1                   /47/0004/00351100080002033ad200
2                   /47/0004/0035150008000203892300
Destination Unreachable
```

This message indicates that the router /47/0004/0035150008000203892300, the last router attempting to reach the destination, did not have a route and returned an error response.

- Access the last router to respond by entering the TELnet command.

In the example in step 3, the last router to respond is router 2, which has the NSAP address of /47/0004/0035150008000203892300. Using the example, enter:

```
TELnet /47/0004/0035150008000203892300
```

You will receive the following message:

```
N-selector changed to 06, trying /47/0004/0035150008000203892306
Connecting ... connected
Escape character is '^]'
NetLogin:
```

This message indicates that you have successfully connected to the last router attempting to route to the destination.

- Find the next-hop router in the path toward the destination in the Level 2 Routing Table (AreaAddress /47/0004/00351300) by entering:

```
SHow -ISIS L2Route
```

You will receive this message:

```
Time since last table update: 179 sec. Update count: 13381
----AreaAddress----  -----Metric---  -----Port----  -----IS-----
```

```

/47/0004/00351000      40          1          080002033AD2
/47/0004/00351100      20          1          080002033AD2
/47/0004/00351300      20          1          080002033AD2
/47/0004/00351500       0          -          080002033CC9

```

- 6 Find the network entity title (NET) of the next-hop router in the path toward the destination in the Intermediate System Table by matching the SystemID of the next-hop router in the Level 2 table with the SystemID portion of the Intermediate System Table.

Using the example in step 5, enter:

```
SHow -CLNP IS
```

You will receive the following message:

```

Intermediate System NetworkEntityTitle      SNPA
/47/0004/000351300080002033CC900          %0800020303D6

```

- 7 After you have identified the router that cannot forward your packet, use TELnet to access it, and check the Level 1 Routing Table by entering:

```
SHow -ISIS L1Route
```

You will receive this message:

```

Time since last table update: 133 sec. Update count: 8213
----SystemID----      -----Metric-----      -----Port-----      -----SNPA/IS-----
#080002033CC9          0          -          -
*080002013C37          20          1          IS 080002013C27
Total 0 ES routes, 2 IS routes

```

The SystemID is not in the Level 1 Routing Table, and this table has all of the ESs and ISs for the Area/47/0004/00351300. There is only one other router in this area, and it is not the one you want to reach. For more information, see “Troubleshooting the Configuration” earlier in this chapter.

Getting Statistics To examine the statistics of the OSI router, follow these steps:

- 1 After the router is up and running, examine the CLNP statistics by entering:

```
SHow -SYS STATistics -CLNP
```

- 2 After the router is up and running, examine the ISIS statistics by entering:

```
SHow -SYS STATistics -ISIS
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information on these parameters, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*. For information on interpreting the statistics displays, see the Statistics Displays appendix.

Troubleshooting the Configuration

OSI routing can be difficult to troubleshoot if there is a problem. This section describes some common misconfiguration problems and the basic tools (OPING, OTraceRoute, and TELnet) to solve them.

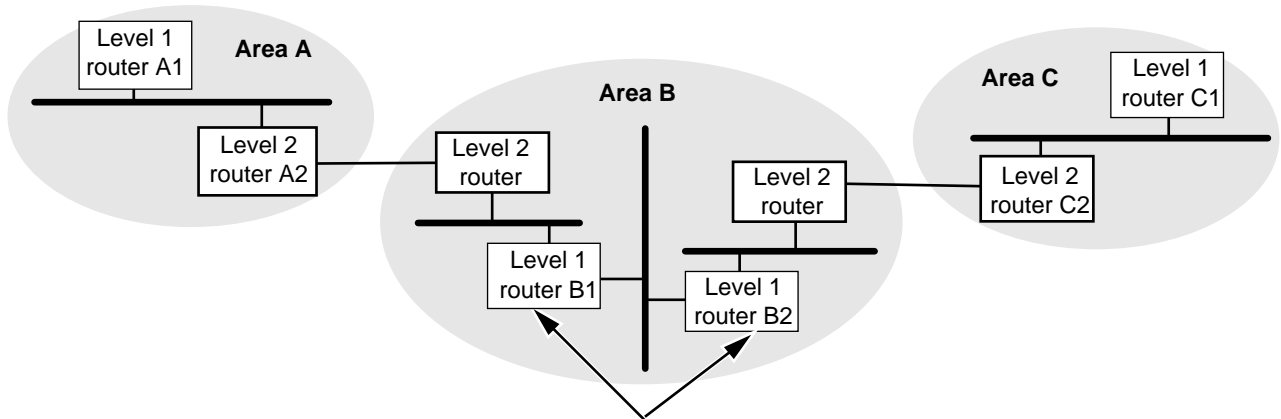
Incomplete Level 2 Backbone

Figure 219 shows an incomplete Level 2 backbone problem that may occur when a transit area has Level 1 routers disrupting the Level 2 path. Area B has broken

the Level 2 backbone. The Level 2 information from Area A cannot be distributed to Area C, and Area A has no Level 2 information from Area C.

The solution to this problem routers B1 and B2 as Level 2 routers with the ISIS parameter mode.

Figure 219 Completing the Level 2 Backbone



Configure Level 1 routers as Level 2 routers with ISIS parameter MODE

Partitioned Area

A partitioned area may occur if multiple routers with the same AreaAddress exist on the network and there is no intra-area route between each pair of these routers. Communication within one partition may succeed, but communication outside the partition may exhibit connectivity problems. Packets that originated from a partition and sent to another area may be delivered without any problem, but packets destined to a system within a partitioned area may be forwarded to the wrong partition.

Another symptom of this problem occurs when some return packets are received and others are not received. This situation exists if multiple routes exist to the partitioned area, and some routes route packets to one partition, while other routes route packets to other partitions.

An area may become partitioned when a link goes down within the area, segmenting the area completely, even though both partitions may still be connected through a Level 2 path through the neighboring areas.

If you suspect a partition, examine for consistency the Level 2 Link State Data for the Domain. Each router indicates its set of area addresses in the Link State PDU identified by the SystemID of the router, followed by the value 00:00.

For example, enter:

```
SHoW -ISIS LinkStateData 080002033ABB:00:00
```

The following display appears:

```
-----ISIS Level 2 Link State Database, Checksum Sum(0008A143)-----
LSP-ID          sequence  remaining  P bit  H bit  attach bit  IS type  data      checksum
                number    lifetime
080002033Abb:00:00  3231    1110      0      0      1           L2       90       17AE(OK)
Area Addresses ==>/47/0004/00351000
```

```
IS neighbors ==> (20 .. ..) 080002033CC9:01
IS neighbors ==> (20 .. ..) 080002013C37:01
IS neighbors ==> (20 .. ..) 080002033ABB:04
IS neighbors ==> (20 .. ..) 080002033ABB:05
IS neighbors ==> (20 .. ..) 080002033ABB:06
IS neighbors ==> (20 .. ..) 080002033ABB:07
IS neighbors ==> (20 .. ..) 080002033ABB:08
```

Multiple Area Addresses

If the routers within an area have more than three area addresses configured, that area may become partitioned. If the extra area addresses have different values, then the algorithm for eliminating extra area addresses may arrive at a different set in different parts of the area and eliminate the most important area addresses. (The algorithm is purely numerical and has no other basis for arriving at the set of three area addresses.)

If you are using more than one AreaAddress for a single area, you can avoid partitioning by configuring the same set of area addresses for every router in the area.

Mismatched Passwords

The IS-IS Protocol has three types of passwords: the interface password (HelloPassWord), the area password (L1PassWord), and the domain password (L2PassWord).

If two routers attached to the same network (LAN or point-to-point) do not have the same HelloPassWord, they will not bring up the adjacency. If you use the L1PassWord to protect against unmanaged routers from becoming attached to your network, then all routers in the same area must be configured with the same password. For the L2PassWord, all Level 2 routers, regardless of the area in which they reside, must have the L2PassWord parameter configured to the same string.

If mismatched passwords exist, see the IS-IS statistics. These statistics show the port on which mismatched passwords occur and the type of failure (Hello, Level 1 or Level 2). The statistics do not identify the misconfigured system; however, you can find the system by examining the IS-IS Adjacency Table and the Link State Database Table.

Customizing the OSI Router

To change the level of routing and configure passwords, follow these steps:

- 1 Determine the level of routing to be used on each port that is enabled for ISIS routing.

Both Level 1 and Level 2 routing are enabled by default. To change level of routing, if necessary, use:

```
SETDefault !<port> -ISIS CONTrol = L2Only
```

The L2Only value under the CONTrol parameter is effective only if the MODE parameter is set to Level2. For complete information on the MODE parameter, see the Configuring OSI Routing chapter in *Reference for Enterprise OS Software*.

For information on Level 1 and Level 2 routing, see "Level 1 Routing" and "Level 2 Routing" later in this chapter.

- 2 Determine whether the port is connected to a stub or transit network. If there are no other routers on a port (for example, a boundary router port), then configure the port as a stub network using:

```
SETDefault !<port> -ISIS CONTrol = Stub
```

- 3 Configure area passwords for all Level 1 routers in the same area.

To configure an area password for the router, use:

```
SETDefault -ISIS L1PassWord = "<password (1-16 characters)>"
```

The Level 1 password prevents routers in another area from accidentally merging into this area.

- 4 Configure passwords for all Level 2 routers using:

```
SETDefault L2PassWord = "<password (1-16 characters)>"
```

Configuring a password prevents other routing domains from learning topology information about this routing domain. It also prevents two routing domains from being accidentally merged into one.

How the OSI Router Works

This section describes the concepts involved in OSI routing activities.

OSI Network Topology

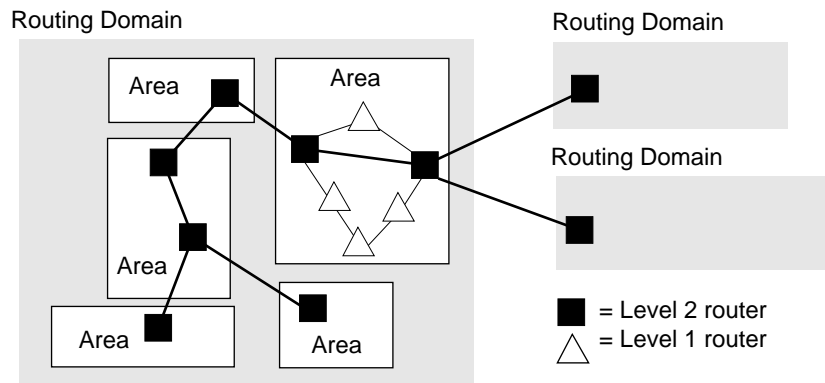
An OSI internetwork is divided into multiple routing domains. The IS-IS routing protocol operates within a routing domain; therefore, it is known as an intra-domain routing protocol.

Inside a routing domain, an OSI network is further partitioned into a two-level hierarchy.

The lower level is called an area. A subset of the IS-IS routing protocol, Level 1, operates within an area. Therefore, Level 1 routing is also known as intra-area routing. The Level 1 routing protocol learns the complete topology in its home area; it does not learn the topology outside of its home area, except for area border routers that can reach other areas. For more information on areas and the Level 1 routing protocol, see "Areas" later in this chapter.

The upper level is called the Level 2 subdomain, which consists of Level 2 routers that connect the areas that make up an OSI routing domain. Another subset of the IS-IS routing protocol, Level 2, operates within the Level 2 subdomain. The Level 2 routing protocol learns the complete topology of the Level 2 subdomain and all areas that it can reach. However, it does not learn the topology of any specific area.

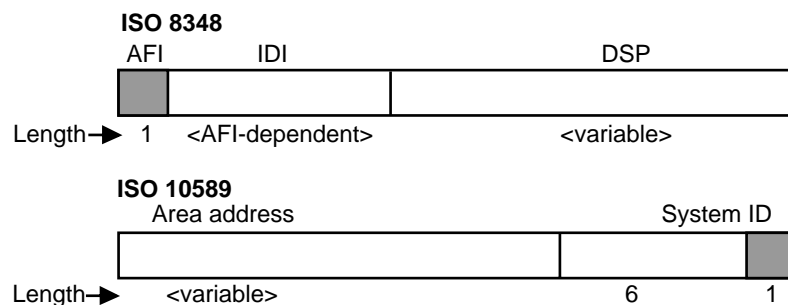
Figure 220 shows the topology of a typical OSI internetwork.

Figure 220 Typical OSI Internetwork Topology**Area Addresses**

The standard for structure and assignment of NSAP addresses is ISO 8348. It defines three fields: the authority and format identifier (AFI), the initial domain identifier (IDI), and the Domain Specific Part (DSP). This structure is useful for creating procedures for assigning unique network service access point (NSAP) addresses, but it is not useful for intradomain routing purposes.

The standard for intradomain routing, IS 10589, views any NSAP address as containing three parts: an area address, a system ID, and an N-selector. The area address identifies an area within the routing domain. The system ID identifies an ES in the area. The N-selector is used by the ES to distinguish between multiple users of the Connectionless Network Service (CLNS), which on the bridge/router includes ISO Transport Class 4 (TP4) and TCP. This structure of the NSAP address is overlaid on the structure of any standard NSAP address as defined by ISO 8348.

Figure 221 shows both structures of the NSAP address.

Figure 221 NSAP Address Structures

The following are examples of area addresses.

Example 1 Suppose your router has the following NSAP address:

/47/0004/0035110008000201345601

The area address is /47/0004/00351100; the ID is 080002013456; the selector is 01.

Example 2 Suppose your router has the following NSAP address:

/49/005308000201345601

The area address is /49/0053; the ID is 080002013456; the selector is 01.

The following area address guidelines must be considered when you set up your OSI network:

- Each router must have at least one area address before IS-IS routing can take place. You can use the 3Com default area address or configure your own. A router can be configured with a maximum of three area addresses.
- Each area should have a globally unique address associated with it. That is, a given area address should be associated with only one area.
- All systems with a given area address must be located in the same area.

Determining Your Own Area Address All bridge/routers shipped from 3Com have the default area address of /49/0053. (In this area address, there is no IDI part for AFI value 49.)

For networks that are not going to be interconnected with other routing domains, you can use the default AFI value 49. The DSP prefix, 0053, can be reassigned with a new value for each different area. For networks such as these, there is room for 65,536 area addresses.

However, 3Com recommends that each installation acquire its own NSAP address block from a registration authority and manage the area addresses from that block. For information on registration authorities and how you can obtain registration information, see the NSAP and PSAP Addressing appendix.

To set an area address for each router, use the AreaAddress parameter. For complete information on this parameter, see the ISIS Service Parameters chapter in *Reference for Enterprise OS Software*.

ID and Selector Values

The ID value is a six-octet field in the NSAP address, as specified in U.S. GOSIP Version 2 DSP format. Because there may be different implementations (which would be incompatible with this implementation) that support different sizes of ID fields, you must ensure that all ISs and ESs use the same ID length within a routing domain.

For all ISs shipped from 3Com, the ID value is automatically extracted from the media access control (MAC) address of the first LAN interface at boot time. You can change this default using the -ISIS SystemID parameter.

The selector is the last octet in the NSAP address. It is used primarily for selecting the transport entity that is to receive a packet. This field is ignored by the IS-IS routing protocol.

Network Entity Title

A router can have multiple area addresses, but it can have only one Network Entity Title (NET). The NET of an IS is computed automatically at boot time. (No user configuration is required.) The NET is computed by taking the area address of the IS and appending the ID value of the IS to it. The selector part is always 00 (see Figure 220).

To display the NET for a particular IS, enter:

```
SHoW -CLNP NetEntityTitle
```

The NET is used primarily for ES-IS and CLNP Protocol operations. Specifically, the NET is used as follows:

- In ISH packets for announcing a router's presence and availability to ESs
- When a router issues CLNP error and redirect protocol data units (PDUs)
- In SourceRoute and RecordRoute options within a CLNP PDU

If there is more than one area address for a router, the NET is computed from the area address made up of the lowest numbers. Since the value of the NET depends on the value of a router's area address, the value of the NET automatically changes to reflect changes to a router's area addresses.

Areas An area is a group of directly interconnected ISs and ESs with the same area address. An area can include anywhere from a single IS up to 100 ISs and up to 2,000 ESs. However, routing in smaller areas runs smoother and more reliably.

3Com recommends that ISs and ESs be grouped into areas based on the following criteria:

- **Departmental function** — for example, manufacturing, engineering, and MIS
- **Administrative or geographic boundaries** — for example, a department, building, campus, or company
- **Level of traffic** — for example, where traffic is heavy and localized, such as a group of workstations and their file servers
- **Reliability of traffic** — for example, where traffic is unreliable and prone to errors, such as a test lab

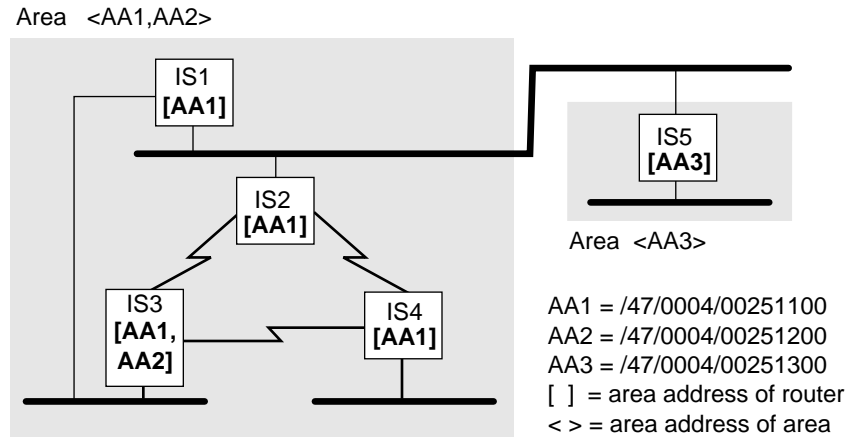
Level 1 Routing

A router configured as a Level 1 router learns routes from other Level 1 routers within the same area. Each router sends out a hello packet to other routers on directly attached subnets. These hello packets contain the area addresses of the router that is sending the packet. By comparing received hello packets, routers may decide that they belong to the same area. These routers then form adjacencies with each other, or they can reject forming an adjacency if there are no area addresses in common. If different hello passwords are defined, the routers will not become adjacent.

The boundary of an area is learned dynamically.

Figure 222 shows a network made up of two areas. In this figure, the area address for a router is enclosed in square brackets ([]), and the area address for an area is shown in angle brackets (< >).

Figure 222 Network Made Up of Two Areas



In Figure 222, IS1, IS2, IS3, and IS4 form an area, because they all share the common area addresses ([AA1]). IS5 forms an area of its own (<AA3>). All routers belonging to the same area must be directly interconnected through physical paths. From any router, it should be possible to reach any other router in the same area through intra-area routes (by going through other routers belonging to the same area).

Not all directly connected routers belong to the same area. For example, IS1 and IS5 do not share an identical area address; therefore, they form two distinct areas. These two areas reside on the same subnet.

Once adjacencies are formed and areas are determined, the adjacent routers within an area exchange routing information.

Level 1 Routing Table

To display the Level 1 Routing Table, enter:

```
SHoW -ISIS L1Route
```

The following display is an example of a Level 1 Routing Table:

```
Time since last table update: 554 sec. Update count: 467
-----System ID-----  ----Metric--  -----Port-----  -----SNPA/IS-----
*080002A034A3           60             2                   IS 080002A014AB
#0800020184A8           0              -                   -
*080002A014AB           20             2                   IS 080002A014AB
*080002A01123           40             2                   IS 080002A014AB
080002000A8F0           80             2                   IS 080002A014AB
080002001312F           20             2                   IS 080002A014AB
* indicates an IS # indicates the nearest L2 IS
```

Entries in the Level 1 Routing Table include the following types of information:

- System ID

The Level 1 Routing Table displays all reachable systems within an area. A system can be an ES, an IS (identified by an asterisk [*]), or the closest Level 2 IS (identified by the pound sign [#]).

- Metric

The Level 1 Routing Table displays the total cost associated with reaching a system within an area.

- Port

The Level 1 Routing Table displays the port number of the router through which the destination is reachable. A hyphen (-) indicates that the system is the router itself.

- SNPA/IS

If a destination is directly attached, the Level 1 Routing Table displays the MAC address of the system (identified by SNPA). If a destination is not directly attached, it displays the system ID of the next hop IS, which is one step closer to the destination (identified by IS).

Level 2 Routing

A router configured as a Level 2 router performs the following functions:

- It runs the Level 2 protocol with other Level 2 routers. It learns routes to other areas from other Level 2 routers throughout the Level 2 backbone.
- It continues to run the Level 1 protocol in its home area, and it learns routes from other Level 1 routers.

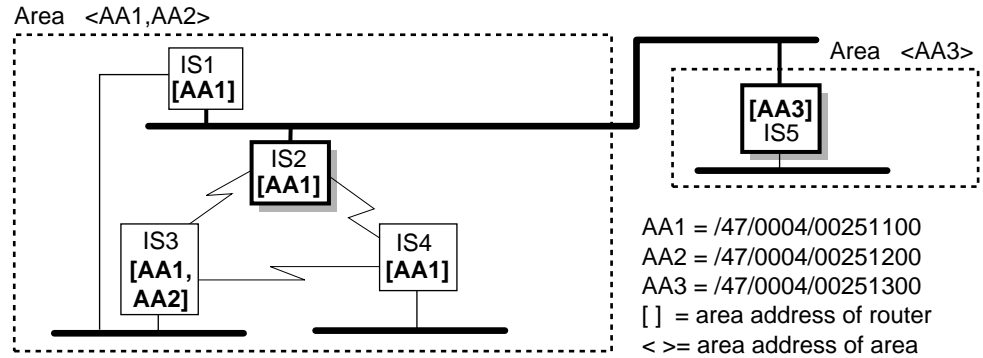
Since the Level 2 protocol runs in parallel with the Level 1 protocol, they do not interfere with each other. As a result, a Level 2 router continues to serve the intra-area traffic for its home area. A nearby ES should not notice a difference in the behavior of this router.

The primary purpose of a Level 2 router is to interconnect disjointed areas into one single routing domain, thus establishing connectivity between areas.

At least one router from each area is selected and configured as a Level 2 router. For example, in Figure 223, if IS2 is chosen to be a Level 2 router for Area <AA1, AA2>, and IS5 is chosen to be a Level 2 router for Area <AA3>, the resulting Level

2 backbone is shown (as indicated by the shadows on the boxes containing IS2 and IS5).

Figure 223 Level 2 Backbone with One Level 2 Router in Each Area



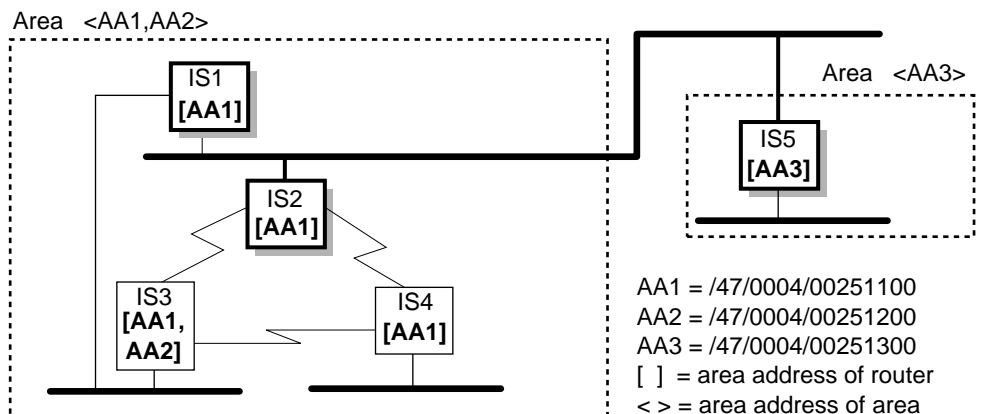
Each Level 2 router belongs to its home area. The Level 2 router summarizes the area address(es) of its home area and announces it to all other Level 2 routers on the Level 2 backbone.

In Figure 223, IS2 announces that it can reach all hosts with Area Addresses AA1 and AA2. With this information, IS5 knows how to reach those hosts. If IS2 had not been configured as a Level 2 router, there would be no way for IS5 to learn the location of Area <AA1, AA2> even though it is directly attached on the same subnet.

You can configure more than one Level 2 router in each area. Figure 224 shows the same topology as in Figure 223 except that IS1 is also configured as a Level 2 router (as indicated by the shadows on the boxes containing IS1, IS2, and IS5).

In Figure 224, Area <AA1, AA2> now has two Level 2 routers, IS1 and IS2, bordering the Level 2 backbone. If one of these routers fails, the other can continue to serve interarea traffic. From the viewpoint of IS5, if it wants to deliver a PDU to Area <AA1, AA2>, it can select either IS1 or IS2. In fact, it can split the load between the two routers. It does not know which router reaches an ES. The detailed topology information within an area is hidden from the Level 2 backbone. All IS5 knows about Area <AA1, AA2> is that it has two area addresses and both IS1 and IS2 can reach it.

Figure 224 Level 2 Backbone with Multiple Level 2 Routers in One Area



All Level 2 routers must be physically interconnected. If a Level 2 router goes down, then the area represented by this router is no longer reachable. A Level 2 backbone should have sufficient redundancy so that the failure of one router or one link does not isolate any area.

Level 2 Routing Table

To display the Level 2 Routing Table, enter:

```
SHow -ISIS L2Route
```

The following display is an example of a Level 2 Routing Table:

```
Time since last table update: 587 sec. Update count: 473
-----Area Address-----  ---Metric--  ----Port---  -----IS-----
/47/0004/00351100           20           1             080002A014AB
/47/0004/00351200           20           2             080002A00949
/47/0004/00351300           0            -             -
/47/0004/00352000           20           2             080002A049B9
```

Entries in the Level 2 Routing Table include the following information:

- Reachable areas
- The Level 2 Routing Table displays all reachable areas within a routing domain.
- Metric
- The Level 2 Routing Table displays the total cost associated with reaching another area within a routing domain. The metric displayed in this table is the total cost of reaching an area border router only. Additional costs may be incurred when traveling from the area border router to the final destination within the area.
- Port
- The Level 2 Routing Table displays the port number of the router through which the next hop IS is reachable. A hyphen indicates it is the router's home area.
- IS
 - The Level 2 Routing Table identifies the next IS that would need to be traversed to reach the destination area. The table displays the system ID of the IS, not the MAC address.

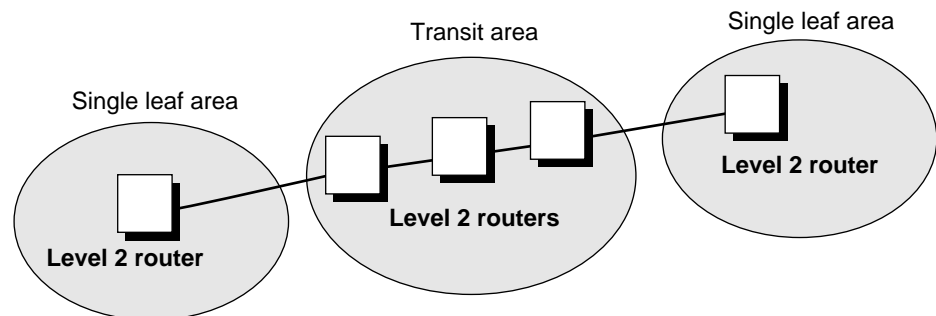
Transit and Leaf Areas

A single leaf area is an area that receives traffic only for itself; it needs only one Level 2 router at the point where it is attached to the neighbor area. As shown in Figure 225, traffic enters each single leaf area and stops in that area.

In contrast, a transit area is an area that receives traffic for both itself and for other areas; it needs Level 2 routers in order to complete the Level 2 backbone. As shown in Figure 225, traffic enters the transit area and can be further routed to the single leaf areas; a transit area interconnects other areas. A leaf area needs only one Level 2 router at the point of attachment to the neighbor area. In order

to complete the backbone, a transit area must contain a *path* of Level 2 routers (see the path of three Level 2 routers within the Transit Area in Figure 219).

Figure 225 Single Leaf and Transit Areas



You must configure the same Level 2 password for each Level 2 router in the same domain.

Metrics and Route Selection

The router running the IS-IS routing protocol selects the path with the lowest total cost to reach its destination. In this case, cost is a user-defined value that measures the capacity of a particular port. A higher value (for example, 50) indicates a higher cost (or a lower capacity). Conversely, a lower value (for example, 10) indicates a lower cost (or a higher capacity).

The total cost to a particular destination is computed by adding the costs of all links toward the destination.

Imagine that there are two routes to a particular destination. Route 1 has a total cost of 100 associated with it; Route 2 has a total cost of 115 associated with it. The router running the IS-IS Protocol will select Route 1, because it has the lowest total cost associated with it.

By default, the cost on all ports has been set to 20, regardless of the underlying network type or speed. These cost values should be adjusted according to your particular situation. For example, a 10 Mbps LAN is preferable to a 64 kbps serial line. In this case, you can set a low cost for the LAN and a higher cost for the serial line.

The L1DefaultMetric and L2DefaultMetric parameters allow you to define the cost associated with using a particular port. For complete information on these parameters, see the ISIS Service Parameters chapter in *Reference for Enterprise OS Software*.

Multipath Routing and Load Splitting

Your topology may contain multiple paths with equal minimum costs toward the same destination.

The OSI router supports multipath routing. It can compute up to four paths toward any destination.

Specifically, for intra-area routing, a Level 1 router can compute multiple paths toward all ESs within the area and toward the closest Level 2 router. If there are multiple Level 2 routers with the same minimum cost, one is randomly selected. For interarea routing, a Level 2 router computes multiple paths toward any area. For interdomain routing, a Level 2 router computes multiple paths toward any domain border router that advertises the longest matching address prefix.

After computing multiple paths toward a destination, the router can then perform load splitting. The router splits the traffic load between the paths on a round-robin basis. Load splitting helps prevent some network segments from being heavily congested, while others are underutilized.

End System Table

An End System Table consists of both dynamic and static entries. The router learns the presence of an ES on the network from the end system hello (ESH) packets. You can also modify the table by adding or deleting ESs.

The following display is example of the End System Table displayed by the SHow -CLNP ES command:

| EndSystem | SNPA | Interface |
|---------------------------------|---------------|-----------|
| /47/0004/001E000108000200E10E01 | %08000200E10E | 2 |

Intermediate System Table

An Intermediate System Table consists of only dynamic entries. The router learns the presence of an IS on the network from the ISH packets. To ensure that the router properly learns the ISs on the network, it is recommended that you not change the default MulticastES and MulticastIS parameter values.

The following is an example of the Intermediate System Table displayed by the SHow -CLNP IS command:

| Intermediate System Network Entity Title | SNPA | Interface |
|--|---------------|-----------|
| /47/0004/001E000108000200999900 | %080002009999 | 2 |
| /49/0053080002A00B7900 | %080002A00B79 | 1 |

User Configurations

Table 53 shows how to change the way the router learns about the network through the ES-IS Protocol. It includes only the parameters that have not been discussed in previous sections. For more information on parameters available in the ESIS Service, see the ESIS Service Parameter chapter in *Reference for Enterprise OS Software*.

Table 53 Configuring the ESIS Parameters

| Parameter | Result |
|---------------------------|--|
| CONTrol: | |
| CheckSum NoCheckSum | Determines whether checksum is used for the ISH PDUs and ESH PDUs. |
| FastConfig NoFastConfig | Determines how fast the router learns about its neighbors. |
| UpdateTime | Determines the interval at which the router sends out ISH PDUs. |

Table 53 Configuring the ESIS Parameters (continued)

| Parameter | Result |
|-----------|--|
| HoldTime | Determines the value of the hold-time field in the ISH PDUs and specifies how long the recipient of the ISH PDUs remembers them. 3Com recommends that it be set to greater than twice the value of UpdateTime. |

Table 54 shows how to change the way the router learns about the network through the IS-IS Protocol. It includes only the parameters that have not been discussed in previous sections. For more information on parameters available in the ISIS Service, see the ISIS Service Parameters chapter in *Reference for Enterprise OS Software*.

Table 54 Configuring the ISIS Parameters

| Parameter | Result |
|------------------|--|
| CsnpTime | Sets frequency at which Complete Sequence Numbers PDUs are transmitted. |
| DISHelloTime | Sets frequency at which hello packets are transmitted by a designated IS. |
| HelloTime | Sets frequency at which hello packets are transmitted by an IS. |
| L1BufferSize | Determines maximum size of Level 1 routing packets sent by an IS. |
| L2BufferSize | Determines maximum size of Level 2 routing packets sent by an IS. |
| L1Multicast | Sets the multicast address that all Level 1 ISs on an Ethernet should transmit hello and routing packets to. |
| L2Multicast | Sets the multicast address that all Level 2 ISs on an Ethernet should transmit hello and routing packets to. |
| LspBroadcastTime | Sets maximum frequency at which routing packets are transmitted on a broadcast network. |
| LspMaxTime | Sets the maximum interval between regenerations of Link State PDUs. |
| LspMinTime | Sets minimum interval between event-driven regenerations of Link State PDUs. |
| LspRtxTime | Sets interval between retransmissions of an update on a point-to-point link. |
| PsnpTime | Sets frequency at which Partial Sequence Numbers PDUs are transmitted. |

Table 55 shows parameters in the CLNP Service that allow you to customize the configuration of your OSI router. It includes only the parameters that have not been discussed in previous sections. For more information on parameters available in the CLNP Service, see the CLNP Service Parameters chapter in *Reference for Enterprise OS Software*.

Table 55 Configuring the CLNP Parameters

| Parameter | Result |
|--------------|--|
| RDgeneration | Determines the frequency at which redirect packets (RD PDUs) are originated by the router. |
| ERgeneration | Determines the frequency at which error packets (ER PDUs) are originated by the router. |

Setting Up Interdomain Routing

A routing domain is usually a single administrative domain, such as a company or a university that runs a single compatible intradomain routing protocol (such as IS-IS). It is composed of one or more (up to several hundred) areas. The areas within the routing domain are interconnected by Level 2 routers. These Level 2 routers make up the Level 2 subdomain, or backbone, within this particular routing domain.

Prerequisites

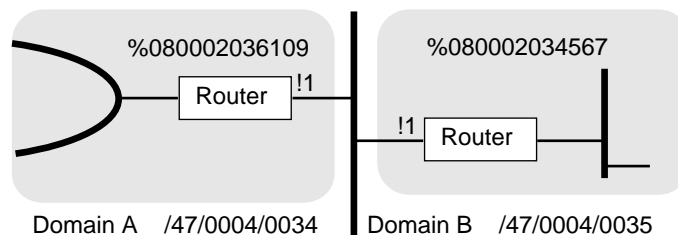
Because interdomain routers are not dynamically learned, you must set up static routes between neighboring Level 2 routers in each routing domain. These routes then are distributed to the other Level 2 routers in the domain through the IS-IS Protocol.

Configure all area addresses with the same initial string of digits, or routing domain Identifier. This string of digits can be used by another domain to create a route to the domain it identifies. A domain built with more than one format of NSAP address results in multiple entries for the various formats. Interdomain routing is based on the longest matching prefix. If a packet contains no matching prefixes in a destination NSAP address, a zero-length default route that matches all NSAP addresses may be used.

Procedure

For an example of setting up static routes between routing domains, see Figure 226.

Figure 226 Two Routing Domains on a Common Ethernet



To set up interdomain routing, follow these steps:

- 1 Isolate the routing domains on each router on the common subnet of the adjoining domains by using the IS-IS HelloPassWord commands:

For example, on the border router of domain A, enter:

```
SETDefault !1 -ISIS HelloPassWord = "Domain-A"
```

On the border router of domain B, enter:

```
SETDefault !1 -ISIS HelloPassWord = "Domain-B"
```

- 2 Configure the PrefixRoute for domain A on the border router from domain B.

For example, on the border router of domain A, enter:

```
ADD !1 -ISIS PrefixRoute /47/0004/0035 %080002034567
```

- 3 Configure the PrefixRoute for domain B on the border router from domain A.

For example, on the border router of domain B, enter:

```
ADD !1 -ISIS PrefixRoute /47/0004/0034 %080002036109
```

- 4 Configure a default route on the border router of domain A to forward to domain B if a matching prefix route cannot be found.

For example, on the border router of domain A, enter:

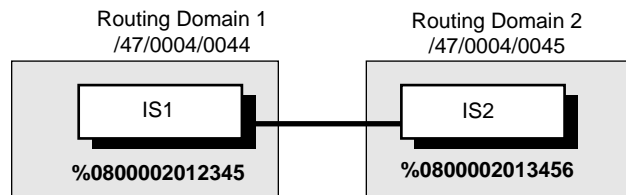
```
ADD !1 -ISIS PrefixRoute Default %080002034567
```

Related Information

Figure 227 illustrates a sample scenario for configuring address prefixes for routing toward other routing domains. Suppose port 1 on IS1 in Routing domain 1 interfaces port 1 on IS2 in Routing domain 2. To set up a static route from routing domain 1 to routing domain 2, enter the following command on IS1 in routing domain 1:

```
ADD !1 -ISIS PrefixRoute /47/0004/0045 %0800002013456
```

Figure 227 Setting Up Interdomain Routing



Conversely, to set up a static route from routing domain 2 to routing domain 1, enter the following command on port 1 of IS2 in routing domain 2:

```
ADD !1 -ISIS PrefixRoute /47/0004/0044 %0800002012345
```

In the above example, note the following:

- Both IS1 and IS2 must be Level 2 routers.
- /47/0004/0044 is the address prefix that summarizes routing domain 1; /47/0004/0045 summarizes routing domain 2. All systems in each domain should have addresses falling under their respective prefixes.
- The link between IS1 and IS2 is a LAN link; therefore, a remote MAC address was specified. You must specify a remote DTE for X.25, an SMDS address for SMDS, or a DLCI number for Frame Relay links.
- If routing domain 2 is a national or regional backbone (meaning that it serves to interconnect many routing domains), it is more appropriate to specify a default route. To set up a default route, for example, on port 1 of IS2, enter:

```
ADD !1 -ISIS PrefixRoute Default %080002013456
```

After you specify the static routes, this information is propagated throughout the Level 2 backbone within the routing domain. All Level 2 routers learn the set of reachable address prefixes and which router can be used to reach that address. If two routers can reach the same address prefix (for example, there are two domain border routers connecting the same external domain), a router selects the domain border router that is closest to it. In this situation, you may also want to configure these routers to perform load splitting. For more information on load splitting, see “Multipath Routing and Load Splitting” earlier in this chapter.

Address Prefixes An address prefix is some number of leading digits of a full NSAP address. It can be as few as two digits or as long as a full NSAP address, whatever is required to uniquely identify another routing domain.

An address prefix points a packet that is being routed between routing domains toward the desired routing domain.

Example Suppose that the Acme Company has been assigned the following NSAP address prefix by the appropriate authority:

/47/0004/0025XXXX

The XXXX field is left for you (the network manager) to assign. You can assume that all hosts with NSAP address prefix /47/0004/0025 reside within the Acme Company and that all hosts within the company have that identical prefix. Assuming that the Acme Company is a single routing domain, then the routing domain can be categorized by address prefix /47/0004/0025.

An address prefix has the following characteristics:

- It can contain an odd number of digits (semi-octets). Examples include /47/0, /47/000, and /47/0004/0. All ranges of AFI values, including both binary and decimal syntaxes, are supported. However, AFI values 50 and 51 are not supported.
- Longer address prefixes take higher precedence over shorter ones. For example, an NSAP address may match multiple address prefixes, as shown:
 - /47
 - /47/0004
 - /47/0004/0035
- Since /47/0004/0035 is the longest address, it is chosen.
- Default routes always have the lowest precedence. They match to all NSAP addresses.

Sometimes a routing domain cannot be assigned a single address prefix. (The routing domain may have been allocated multiple NSAP addresses from different authorities.) Therefore, you must set the -ISIS PrefixRoute parameter for each NSAP address type. The following example illustrates how to set the PrefixRoute parameter for different NSAP address types.

Example Suppose that the AAA Company has merged with the BBB Company to form the CCC Company. Both the AAA and BBB Companies already had large OSI networks. The AAA Company's network used an NSAP addressing scheme based on the U.S. GOSIP version 2 format, while the BBB Company's network used an ANSI-based addressing scheme. If another party (such as a company) needs to communicate with the CCC Company, it must configure its domain border routers with the two address prefixes originally used by the AAA and BBB companies. The following are examples of commands that you can enter on a domain border router to configure it to communicate with the CCC Company over an X.25 PDN:

```
ADD !3 -ISIS PrefixRoute /39/840/543621 #030354321982608
ADD !3 -ISIS PrefixRoute /47/0005/016A9F72 #031354321982608
```

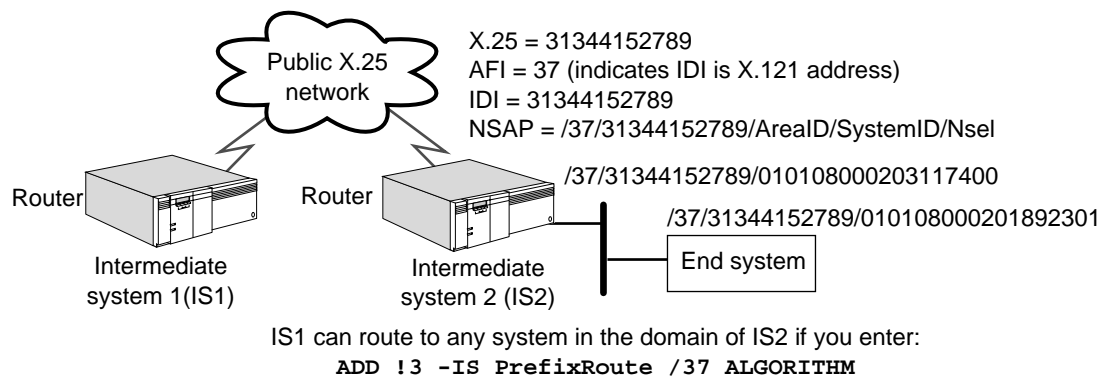
Static Interdomain Routing If your OSI network is attached to a public X.25 or SMDS network, the address of the public network that identifies your router as a

node on the public network also identifies you as an addressing authority according to the standard ISO 8348. Special AFIs for NSAP addresses are formed using the public network address as the IDI, and you can structure the DSP in any way, provided that the last seven octets are the system ID and the N-selector.

If this is the case, and multiple sites are interconnected using the same method for their NSAP address assignments (reachable directly over the same public network), then interdomain routing can be accomplished using an algorithm. The algorithm used extracts the public address from the NSAP address and forwards the CLNP packet on the public network using that extracted address as the SNPA of the next hop router.

This form of address and method of determining the next-hop media address across the public network lets you build extremely large OSI networks.

Figure 228 Domain Addressing Based on X.25 Attachment Address



Interdomain Routing Table There are two forms of the Interdomain Routing Table.

The first form displays a collective listing of all routing domains that can be reached from a particular routing domain. To display this routing table, enter:

SHow -ISIS PrefixRoute

The following display shows a typical Interdomain Routing Table:

| -NSAPAddress Prefix- | --Metric-- | --Port-- | -----IS/SNPA----- |
|----------------------|------------|----------|---------------------|
| /47/0004/00352 | 20 | 1 | SNPA %0800002A00AB6 |
| /47/0004/0035 | 20 | 1 | SNPA %0800002A00B92 |
| /47/0004 | 20 | 1 | SNPA %0800002013C37 |
| Default | 20 | 2 | IS %0800002019876 |

Entries in the Interdomain Routing Table include the following information:

- NSAP address prefix

This routing table displays reachable NSAP address prefixes.

- Metric

This routing table displays the total cost associated with reaching a particular routing domain. The metric displayed in this table is the total cost of reaching a domain border router only. Additional costs may be incurred when traveling from the domain border router to the final destination.

- Port

- This routing table displays the port number of the router through which the destination routing domain is reachable.

- Next Hop (IS/SNPA)

If the external domain is directly reachable, this routing table displays the MAC, SMDS, or data terminal equipment (DTE) address (or data link connection identifier (DLCI)) that can be used to reach this domain (identified by SNPA). This information is displayed when the router itself is the domain border router and has been configured with address prefix information. Otherwise, the routing table displays the next hop IS, which is one step closer to the domain border router that has been configured with address prefix information (identified by IS).

The second form displays the static routes you configured on a particular router. To display these static routes, enter:

SHowDefault -ISIS PrefixRoute

The following display is a table of the static routes:

| ---NSAPAddress Prefix---- | --Port-- | -----SNPA---- | ---Status-- |
|---------------------------|----------|----------------|-------------|
| /47/0004/00352 | 1 | %0800002A00AB6 | Active |
| /47/0004/0035 | 1 | %0800002A00B92 | Active |
| /47/0004 | 1 | %0800002A13C37 | Active |

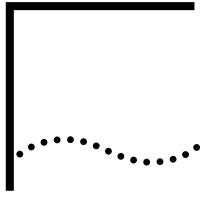
Entries in this form of the Interdomain Routing Table include the following types of information:

- NSAP (Network Service Access Point) address prefix
This routing table displays reachable NSAP address prefixes.
- Port
This routing table displays the port number of the router through which the destination routing domain is reachable.
- SNPA (Subnetwork Point of Attachment)
This routing table displays the MAC, SMDS, or DTE address (or DLCI) that can be used to reach an external domain.
- STATUS
This routing table displays the status of the NSAP address prefix. An "active" status indicates that the address prefix is operational. An "idle" state indicates that the address prefix is not in service. The address prefix may be in the idle state if the port associated with the prefix is down, the router has not been configured to perform Level 2 routing, or the SNPA syntax is rejected by the lower layers (for example, a DTE address may be specified on a Frame Relay port or a MAC address may be specified on an X.25 port).

Integrated IS-IS for IP and Dual IP/OSI Mode

Integrated IS-IS is a protocol that provides integrated OSI-type routing for IP and OSI environments. It is the IP extension added to the original OSI IS-IS Protocol. Integrated IS-IS routing simplifies network topology, reduces network management complexity, and reduces routing traffic overhead.

To configure Integrated IS-IS for IP and dual IP/OSI environments, see the Configuring IP Routing chapter.



CONFIGURING VINES ROUTING

This chapter describes the procedures for configuring your system to perform VINES IP routing. It also describes how the router works and gives guidelines for operating, managing, and troubleshooting the router.



For conceptual information, see “How the VINES Router Works” later in this chapter.

Setting Up a Basic VINES Router

VINES network numbers and addresses are not user-configurable in the same way that other routing protocols are. The router automatically assigns its own VINES network address, enabling the router to communicate with VINES servers once VINES routing is enabled. This VINES network address is 32 bits long and consists of two parts. The first part of the network number is a specific vendor code that Banyan Systems has reserved for 3Com. This vendor code, which starts with hex 302 or hex 303, is composed of the 11 most significant bits of the 32-bit network address. The remaining 21 bits of the network number contain the 21 least significant bits of the router MAC address.

Configuring for Local Area Networks and Point-to-Point Protocol Links

The procedure in this section explains how to enable VINES routing and set up parameters on LAN ports and Point-to-Point Protocol (PPP) links where no VINES servers are available.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Procedure

To configure the system to perform basic VINES routing, follow these steps:

- 1 Enable VINES routing using:

```
SETDefault !<port> -VIP CONTROL = Route
```

- 2 Enable ARP on those ports where no VINES servers are available using:

```
SETDefault !<port> -VIP CONTROL = Arp
```

The specified port will now respond to Address Resolution Protocol (ARP) query and ARP assignment requests.

- 3 Forward VINES broadcast packets when the nearest VINES server is more than one hop away from a VINES client using:

```
SETDefault !<port> -VIP CONTROL = NoServer
```

- 4 Select the packet encapsulation format for each Ethernet interface using:

```
SETDefault !<port> -VIP HeaderFormat = [Ethernet | Ieee | Snap]
```

and specifying either Ethernet, leee, or Snap.

For more information, see “HeaderFormat” in the VIP Service Parameters chapter in *Reference for Enterprise OS Software*.

- 5 Control whether the router forwards broadcast packets over a port where packet charges are enforced using:

```
SETDefault !<port> -VIP CONTROL = PktChrg
```

This value prevents the router from forwarding broadcast packets received from other reachable nodes and servers, unless the class subfield bit in the Transport Control field is set appropriately.

- 6 Verify the VINES configuration by entering:

```
SHoW -VIP CONFIguration
```

The router displays the configuration information for active VINES ports only. If there is no active port, it prompts you to enable VINES routing. To display the default configuration, enter:

```
SHoW !* -VIP CONFIguration
```

To complete the procedure for PPP links, see the Configuring Wide Area Networking Using PPP chapter.

Configuring for Wide Area Networks

You can configure the VINES router to perform routing over wide area network ports using Point-to-Point Protocol (PPP), Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), Switched Multimegabit Data Service (SMDS), X.25, and Integrated Services Digital Network (ISDN).

Routing VINES over Frame Relay, ATM DXI, and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to route VINES over Frame Relay, ATM DXI, or X.25 in a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. For complete information on configuring VINES routing over Frame Relay, ATM DXI, or X.25, including a discussion on fully meshed, partially meshed, and nonmeshed topologies and virtual ports, see the Configuring Wide Area Networking Using Frame Relay chapter, the Configuring Wide Area Networking Using the ATM DXI chapter, and the Configuring Wide Area Networking Using X.25 chapter, respectively. For information on the number of virtual ports supported per platform, see Table 11 in the Configuring Advanced Ports and Paths chapter.

Routing VINES over SMDS is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your VINES router to perform routing over SMDS, see the Configuring Wide Area Networking Using SMDS chapter.

To configure your VINES router to perform routing over PPP, see the Configuring Wide Area Networking Using PPP chapter. For information on wide area networking using ISDN, see the Configuring Wide Area Networking Using ISDN chapter.

Verifying the Configuration

After you have configured the basic VINES router, check to see whether it can route packets properly. Examine the VINES routing and neighbor tables, and send packets from one network to another to see if they are properly forwarded.

Verifying Procedure

Before you use the router for interconnecting networks, follow these steps to verify the router configuration:

- 1 Check the router path configuration by entering:

```
SHow -PATH CONFIguration
```

- 2 Check the router port configuration by entering:

```
SHow -PORT CONFIguration
```

- 3 Verify the VIP Service configuration by entering:

```
SHow -VIP CONFIguration
```

This command displays VINES configuration information and other related data.

- 4 Check the status of each port on the VINES router by entering:

```
SHow -VIP STATUS
```

The SHow -VIP STATUS command shows the status of each port, either Up or Down.

- 5 Examine the routing table to see if the destination networks are reachable by entering:

```
SHow -VIP AllRoutes
```

This command displays all known routes in the VINES Routing Table.

- 6 Display all known neighbors in the neighbor table by entering:

```
SHow -VIP Neighbor
```

Getting Statistics

To check statistics for the VINES router, enter:

```
SHow -SYS STATistics -VIP
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*. For information on interpreting the statistics displays, see the Statistics Displays appendix.

Checking Reachability

You can use the VPing command to check if a specific server or router is reachable or alive. If the target server is not reachable, try reaching the intermediate routers and locate the source of the problem.

To ping a VINES server, use:

```
VPing <server address>
```

The following message appears if the target server is alive:

```
Pinging... 2901599 is alive
```



The target server must be running VINES 5.0 or greater or 3Com Enterprise OS software version 6.2 or greater.

For more information on the VPing command, see the Commands chapter in *Reference for Enterprise OS Software*.

Troubleshooting the Configuration

If you are unable to make connections to other networks after setting up the router, review the following troubleshooting procedure. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier or 3Com for assistance.

Procedure

To troubleshoot the VINES configuration, follow these steps:

- 1 Check that all cables on all routers in a specific path in the routing table are properly connected and that the routers are properly installed.

For installation instructions, see the installation guide provided with your bridge/router.

- 2 Check the VIP CONTrol status by entering:

```
SHoW -VIP CONTrol
```

The router displays the current values for the CONTrol parameter. If one of these values is set to NoRoute, enable the VINES router using:

```
SETDefault !<port> -VIP CONTrol = Route
```

- 3 Check the VINES network status by entering:

```
SHoW -VIP STATUS
```

Look at the status of the networks. All networks should be in the Up state. If any one is in the Down state, check to make sure that all PORT and PATH parameters are configured appropriately.

- 4 Check whether a specific neighbor is up and running by entering:

```
SHoW -VIP Neighbor
```

If a neighbor is up and running on the network, it will appear in the neighbor table.

- 5 Check whether the network you are trying to reach is in the VINES Routing Table by entering:

```
SHoW -VIP AllRoutes
```

The VINES router displays the routing table entries. From the table, you can determine which path is being used. Examine the entries to make sure that a route in the table is taking the appropriate path.

If the entry in the table has a hop number of 65535 (hex FFFF), the network is unreachable at the present time. Wait several minutes and enter the SHoW -VIP AllRoutes command again.

- 6 Display statistics for the VIP Service by entering:

```
SHoW -SYS STATistics -VIP
```

For information on interpreting the statistics displays, see the Statistics Displays appendix.

Customizing the VINES Router

You can customize the VINES router configuration by assigning a name to the local VINES router and assigning symbolic names to neighbors in your VINES network for tracking purposes.

To assign a name to the VINES router network number, use the SETDefault -VIP RtrName command. You can rename the router to any string up to 16 characters, but the name must be unique in the VINES network.

For example, to assign the name "3Com.Engr" to the router, enter:

```
SETDefault -VIP RtrName = "3Com.Engr"
```

This router name is used when the router responds to VINES Security Service requests, which enforce network security and authentication. The service uses the router name for user ID authentication to determine whether the client from where the user is logging in is a physical neighbor and should be permitted access to the network. The router name is also used when the router responds to service statistics requests from clients invoking the WHATZ command.

To assign symbolic names to other VINES servers on your network, enter the ADD -VIP SymbolicNames command. Adding symbolic names to VINES servers can help you keep track of other VINES servers when you display the VINES neighbor and routing tables. You assign the symbolic name to the VINES network number (which must be entered in hexadecimal form).

For example, to assign the name "Finance.2ndFloor" to a VINES server with the network number 002c465f (hexadecimal), enter:

```
ADD -VIP SymbolicNames 002c465f "Finance.2ndFloor"
```

You can assign up to 128 symbolic names, and each symbolic name can be up to 15 characters long. Symbolic names are only displayed when you specifically request the symbolic name option with the SHow -VIP AllRoutes and SHow -VIP Neighbor commands.



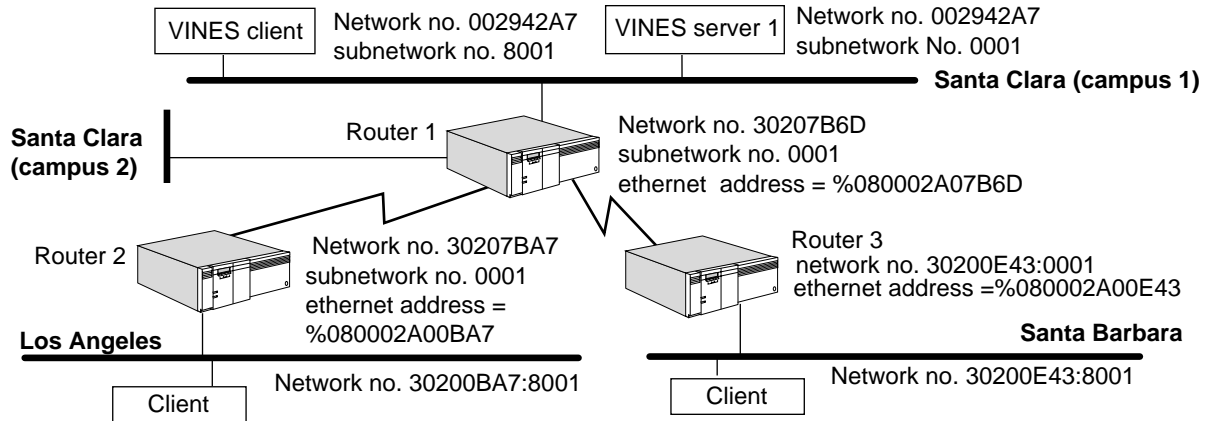
Names configured with the RtrName and SymbolicNames parameters have no relationship to Banyan VINES StreetTalk names, and will not be advertised. The NETBuilder bridge/router does not support StreetTalk name server requests.

How the VINES Router Works

VINES networks are configured automatically on each port, and the configuration is transparent to the user. The port can be a local Ethernet port or a serial line port for a wide area network, such as a point-to-point link or an X.25 link.

Figure 229 is an example showing a wide area router connecting two local Ethernet networks (Santa Clara) to two wide area networks (Los Angeles and Santa Barbara).

Figure 229 Wide Area Router Connecting Four VINES Networks



All 3Com router network numbers start with hex 302 or hex 303. As shown in Figure 229, all routers and servers have unique serial numbers, which are the same as the network numbers. Their subnetwork numbers are always 0001. These servers and routers assign unique network numbers and subnetwork numbers to the client nodes. Client subnetwork numbers can be any number from hex 8000 through hex FFFE. One physical network can have as many logical network numbers as the servers and routers (See Figure 229).

A router must check its routing table to determine where to route a packet. If the destination is one of the neighbors, the router can send it directly to the neighbor. If the destination is not a neighbor, the router must route the packet to another router (called a "gateway") that is closer to the destination. The route to a remote network can be dynamically learned through routing protocols, such as the Routing Table Protocol (RTP) for VINES.

Routing Tables

Two tables are used in VINES routing: the VINES Routing Table and the VINES Neighbor Table.

VINES Routing Table

This table displays all known routes in the routing table. To display the VINES Routing Table, enter the `SHoW -VIP ALLRoutes` command.

The following display is an example of the default routing table:

```
-----VINES Routing Table-----
Port  NET          Gateway      Metric  Port  NET          Gateway  Metric
5     807600533     807600533   45     1     2903035     2903035   2
Total route(s) displayed:2
```

You can also display the routing table in both hex or symbolic formats. To display the routing table in hex format, enter:

SHoW -VIP AllRoutes Hex

To display the routing table in symbolic format, enter:

SHow -VIP AllRoutes Sym

The VINES Routing Table provides the following information:

- Port
- The port number of the router through which the destination is available.
- NET
- This is a logical network number learned dynamically through its neighboring routers or servers.
- Gateway
- The VIP address of the gateway to which a router must send a packet before the packet can be routed to the destination.
- Metric
- The metric for a particular interface. The metric is automatically calculated, and is based on baud rate.

- Status

Indicates the status of the route as follows:

- Up Route is up and usable.
- Dn Route is down and soon to be purged.
- Ch Entry has been recently updated and must be included in the next RTP updates across permanent links.
- Hd1 Route is in the first hold-down period and identifies a network whose unreachable state was recently updated, but not verified.
- Hd2 Route is in the second hold-down period and indicates the unreachable state has been confirmed and it can now be advertised.

The ROUTE status is only displayed when you display the routing table in symbolic or hex format.

VINES Neighbor Table

This table displays all known neighbors in the neighbor table. To display the VINES Neighbor Table, enter:

SHow -VIP Neighbor

The following display is an example of the neighbor table:

```
-----VINES Neighbor Table-----
Port   NETnumber   Media Address   Metric   HdrFmt   Status
1      2903035    %02608CA1B088   2        ETH      Svr/
5      807600533   PPP             45       PPP      Svr/Perm
```

The VINES Neighbor Table provides the following information:

- Port

Identifies the port number of the router through which the destination is available.

- NETnumber

If the neighbor is a service node or a router, it has a unique 32-bit network number. The network number is the serial number of the service node or the router. Each service node or router has 0001 for its subnetwork number. If a neighbor is a client node, it gets its network number and subnetwork number from a service node or a router. Subnetwork numbers range from hex 8000 through hex FFFE.

- Media Address

While network numbers and subnetwork numbers are the logical network numbers of a node, media address represents the underlying data link layer address, such as Ethernet address, X.25 address, or Frame Relay DLCI.

- Metric

Indicates the metric (in 200 millisecond increments).

- Header Format

Indicates whether Ethernet, IEEE, or SNAP packet encapsulation is being used.

- Status

Indicates the status of the neighbor as follows:

Svr Neighbor is a server or a router.
 Clnt Neighbor is a client.
 Perm Neighbor is a permanent, will not age out. Any neighbor learned over a serial line is considered permanent.
 IP Neighbor is learned through IP.
 Redir Neighbor is in the process of RTP redirect.

For each destination address, the router supports only one route.

Routing Selection The VINES router keeps in its routing table only one network number per destination. It does not support backup routes.

Deleting Routes Because VINES does not allow for static route configuration, there is no DElete command that deletes individual routes one at a time. You can delete all the entries by flushing them.

VINES Routing Table entries and neighbor table entries age out if no updates are received for about 9 minutes, which is six times the value of the user-configured UpdateTime parameter. The default value for the UpdateTime parameter is 90 seconds.

To remove all dynamic routes from the VINES Routing Table, enter:

FLush -VIP AllRoutes

This command simultaneously removes all entries from the VINES Neighbor Table so that the two tables remain consistent.

Learning Routes Every time the router learns a route change for a network, or every 90 seconds (by default), it uses broadcast packets to report the following types of information to its neighboring gateways:

- The networks it can reach (4 bytes)
- The metric or cost associated with each network it can reach (2 bytes)

You can configure the UpdateTime parameter in the VINES Service to change the interval at which the router broadcasts routing update packets (RTPs).

Network Reachability, Split Horizon, and UpdateTime The types of networks that are considered "reachable" when a router broadcasts its RTP update packets are as follows:

- A directly connected network
- All dynamic routes learned through RTP in the routing table

To prevent endless routing loops caused by including routes in the updates sent back to the same gateway from which the routes were originally learned, a preventive measure known as *split horizon* is used. To achieve split horizon, the router does not include those routes learned from that interface when it generates RTP update packets to an interface. For example, when a network is learned from a neighbor on port 1, this network will not be included in any updates to port 1 to prevent mutual deception.



VINES servers currently do not support split horizon.

The UpdateTime parameter changes the frequency at which the router sends update packets. The UpdateTime parameter specifies the time interval by which

the router sends its routing table updates. For networks that seldom experience topology changes, the interval time can be set higher than the default value to reduce the amount of network traffic. For networks that often experience topology changes, this value can be set lower than the default value.



The lower you set the UpdateTime value, the more data traffic will be generated on the network; increased traffic can degrade network performance.

Banyan VINES Client/Server Support

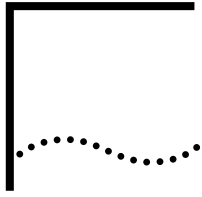
The 3Com VINES router supports a subset of the VINES Protocol suite, such as VINES Internet Protocol (VINES IP), the RTP, Address Resolution Protocol (ARP), and the Internet Control Protocol (ICP). When the VINES router receives broadcast packets, it pays special attention to ICP packets by selectively propagating VINES StreetTalk packets (for the VINES Directory Service), Time Synch Service packets, and VINES Security Service. However, the VINES router does not participate in any other VINES Directory Service.

3Com VINES routers are preassigned with a unique 32-bit network number and a subnetwork number of 0x0001. However, a client must obtain its VINES Internet address from its router or server using the VINES ARP. After a client boots up, it broadcasts an ARP Query Request seeking a response from a server or a router. Any neighbor server or router with the ARP Service enabled responds with an ARP Query Response. Two different versions of VINES ARP are available: sequenced ARP and non-sequenced ARP. All VINES servers and clients running Banyan VINES software previous to version 5.50 use non-sequenced ARP, while servers and clients running VINES software version 5.50 and later use sequenced ARP. For the two types to interoperate, nodes that support sequenced ARP also support non-sequenced ARP. For example, a client node that runs VINES 5.50 can use a VINES 5.0 server if no VINES 5.50 servers are available, and a server that runs VINES 5.50 can provide an ARP Service to a VINES 5.0 client node.

This version of the 3Com VINES router does not support sequenced ARP. The 3Com VINES router uses the RTP to exchange routing information with servers or routers, and to maintain the topology information in the routing table. When routing data packets, the 3Com VINES router makes routing decisions based on the routing database. If the final destination of a packet is a neighbor, the router will send the packet to the neighbor directly. Otherwise, it will send the packet to the next router toward the final destination. Each RTP update packet contains a list of all the networks known to the router and metric for each network.

Two versions of RTP are available: sequenced RTP and non-sequenced RTP. All VINES servers and clients running Banyan VINES software previous to version 5.50 use non-sequenced RTP, while servers and clients running VINES software version 5.50 and later use sequenced RTP. For interoperability, routers that support sequenced RTP also support non-sequenced RTP for backward compatibility. This version of the 3Com VINES router does not support sequenced RTP.

The 3Com VINES router provides support for RTP Redirect. When a unicast packet has to be forwarded on the same port on which it was received and the RTP Redirect bit is set, 3Com routers generate an RTP Redirect packet to inform the last forwarding router or server of a better path to the given destination. The advantage of RTP Redirect is that an unnecessary extra hop can be reduced.



CONFIGURING XNS ROUTING

This chapter describes the procedures for configuring your system to perform Xerox Network Systems (XNS) routing. It also describes how the router works and gives guidelines for operating, managing, and troubleshooting it.



For conceptual information, see "How the XNS Router Works" later in this chapter.

Setting Up a Basic XNS Router

The procedure in this section describes the minimum steps required to enable your system to route XNS packets. Depending on your network requirements, you can use the default values of the parameters, or you can further configure the router according to later sections in this chapter.

The parameters in the IDP and RIPXNS Services enable XNS routing functions.

Configuring for Local Area Networks and Point-to-Point Protocol Links

When setting up the basic XNS router, you first configure the router for LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Procedure

To set up the router for XNS routing, follow these steps:

- 1 Enable XNS routing by entering:

```
SETDefault -IDP CONTROL = Route
```

In addition, you can configure the `-IDP CONTROL` parameter to provide error checking with the `Checksum | NoChecksum` value. Checksum provides a high degree of reliability in detecting bad data sent over the network. If Checksum is enabled, a router verifies the IDP checksum of a packet before it forwards the packet. The cost of this service, however, is lower network performance. The default value is `NoChecksum`.

- 2 Configure XNS network numbers on each port connected (local interface or serial line interface) using:

```
SETDefault !<port> -IDP NETnumber = &<number>(0-FFFFFFFE)
```

Valid network numbers consist of up to eight hexadecimal digits in the range `&0` to `&FFFFFFFE`. The network number `&FFFFFFF` is reserved. Use network number `&0` to delete a previously assigned network number. You do not have to specify leading zeros in the network number.

Repeat this step for the other port(s). Each enabled port on a router must be assigned a different network number.

- 3 Verify the XNS configuration by entering:

SHoW -IDP CONFIguration

The router displays the IDP configuration information. If the CONTRol parameter is not set to route, or the NETnumbers are incorrect, repeat steps 1 and 2.

- 4 Begin routing table information exchanges with other routers that interface with a port using:

```
SETDefault !<port> -RIPXNS CONTRol = Enabled
```

- 5 Repeat step 4 for each port being used for XNS routing.

After you have completed this procedure, dynamic XNS routing begins over the configured ports. To complete the configuration for PPP links, see the Configuring Wide Area Networking Using PPP chapter.

For more information on dynamic and static routes, see "Customizing the XNS Router" later in this chapter.

Configuring for Wide Area Networks

XNS routing over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to route XNS over a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. For complete information on configuring XNS routing over Frame Relay, ATM DXI, or X.25, including a discussion of fully meshed, partially meshed, and nonmeshed topologies and virtual ports, see the Configuring Wide Area Networking Using Frame Relay chapter, the Configuring Wide Area Networking Using the ATM DXI chapter, and the Configuring Wide Area Networking Using X.25 chapter, respectively. For information on the number of virtual ports supported per platform, see Table 11 in the Configuring Advanced Ports and Paths chapter.

Routing XNS over Switched Multimegabit Data Service (SMDS) is supported over fully meshed and nonmeshed topologies (nonmeshed topologies require virtual ports). In addition, SMDS virtual ports are supported and can be used for traffic separation and various filtering of by assigning groups of nodes to different virtual ports. For more information, see the Configuring Wide Area Networking Using SMDS chapter.

To configure your XNS router to perform routing over PPP, see the Configuring Wide Area Networking Using PPP chapter. For more information on wide area networking using Integrated Services Digital Network (ISDN), see the Configuring Wide Area Networking Using ISDN chapter.

Verifying the Configuration

After you have configured the basic XNS router, you should verify the configuration to see if you can reach other XNS hosts.

Before you use the router for interconnecting networks, verify the router configuration by following these steps:

- 1 Check the router path configuration by entering:

SHoW -PATH CONFIguration

- 2 Check the router port configuration by entering:

```
SHoW -PORT CONFIguration
```

- 3 Examine the IDP Service configuration by entering:

```
SHoW -IDP CONFIguration
```

This command displays configuration information specific to the IDP Service parameters for each port that you have configured with a network number.

- 4 Examine the RIPXNS Service configuration by entering:

```
SHoW -RIPXNS CONFIguration
```

This command displays configuration information specific to the RIPXNS Service parameters for each port that you have configured with a network number.

- 5 Check the state of all networks assigned to the ports of a router by entering:

```
SHoW -IDP NETnumber
```

This command displays the network number assigned to each port on this router and the state that each network is in. All networks should be in the UP state. If any one is in the DOWN state, check to make sure that all PORT and PATH parameters are configured correctly.

- 6 Check the XNS Routing Table to see if all the networks are reachable by entering:

```
SHoW -IDP AllRoutes Long
```

This command displays all known routes, both dynamic and static, in the XNS Routing Table.

- 7 Make a connection from a host on one attached network to a host on another network to see if packets can be routed across the router.

You can also test the connectivity between routers by using the REMote command.

Figure 230 shows four Ethernet networks connected by routers A, B, and C.

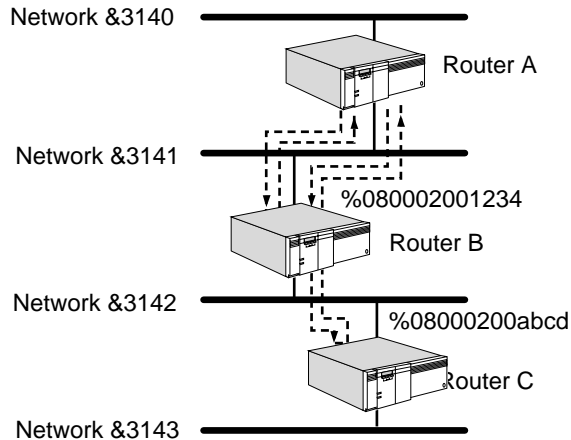
To check the connectivity between router A and router B, on router A enter:

```
REMote &3141%080002001234
```

To check the connectivity between router A and router C, on router A enter:

```
REMote &3142%08000200abcd
```

After you enter the REMote command, the remote prompt (Remote:) appears. At the Remote prompt, enter any command available on the device to which you remote (Routers B or C); for example, SHoW -SYS VERSion or SHoW -SYS ADDRess. A response from Routers B or C indicates successful communication between respective routers.

Figure 230 Checking Connectivity between Routers

Getting Statistics To display statistics for the IDP Service, enter:

```
SHow -SYS STATistics -IDP
```

To display statistics for the RIPXNS Service, enter:

```
SHow -SYS STATistics -RIPXNS
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information on these parameters, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*. For information on interpreting the statistics displays, see the Statistics Displays appendix.

Troubleshooting the Configuration

If you are unable to make connections to other networks after setting up the router, review the following troubleshooting procedure. This procedure can help correct problems in making single-hop (involving one router) and multiple-hop (involving more than one router) connections.

To troubleshoot the basic XNS router configuration, follow these steps:

- 1 Check that all cables on all routers in a specific path in the routing table are properly connected and that the routers are properly installed.

For installation instructions, see the installation guide provided with your bridge/router.

- 2 Check the -IDP NETnumber and the network status by entering:

```
SHow -IDP NETnumber
```

Look at the status of the networks. All configured networks should be in the UP state. If any one is in the DOWN state, check that all PORT and PATH parameters are correctly configured.

Look at the current network configuration. If no network is configured on the specific port, use the SETDefault -IDP NETnumber command to add a proper network number to that port.

- 3 Check the values of -RIPXNS CONTROL parameter by entering:

```
SHow -RIPXNS CONTrol
```

The router displays the current values for the CONTROL parameter.

- 4 Check whether the network you are trying to reach is in the XNS Routing Table by entering:

SHow -IDP AllRoutes

To verify single route reachability, you can specify a network number and enter:

SHow -IDP AllRoutes <NETnumber>

For more information on checking the routing table, see "Displaying Routing Information" later in this chapter.

Customizing the XNS Router

After you set up and check the router according to instructions in the previous sections, you are ready to customize the XNS router by configuring specific routes, which includes the following steps:

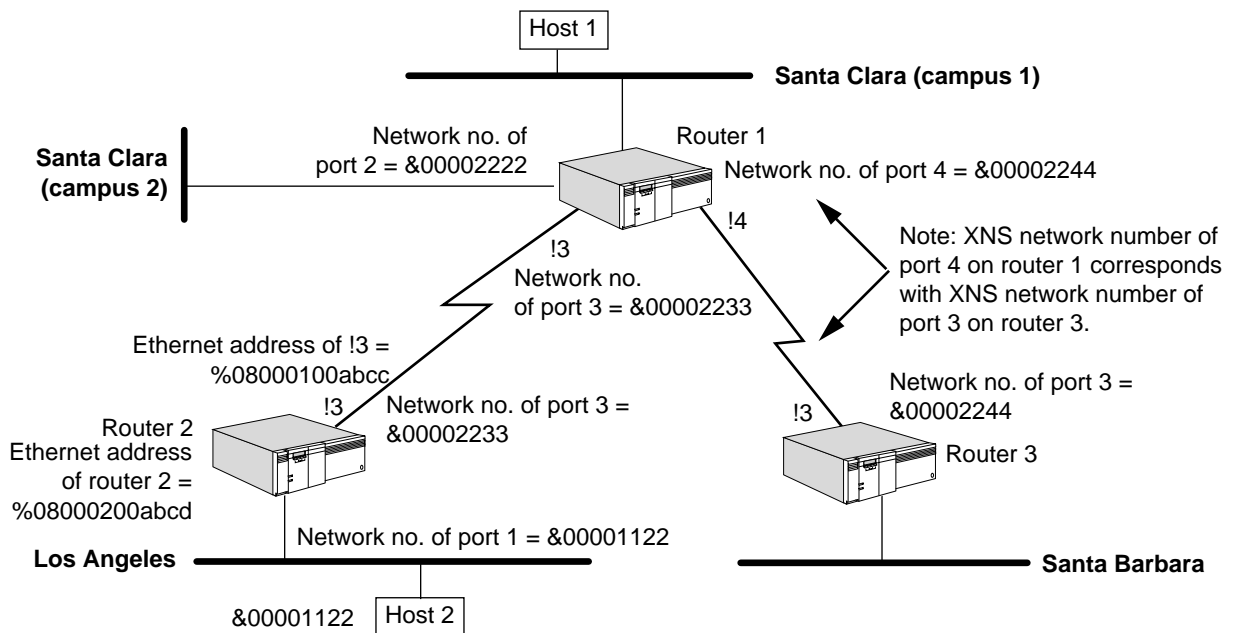
- Determining network routes dynamically and statically
- Making routing decisions (that is, determining whether a packet destination is on an attached network or a reachable remote network and determining how to reach the destination if multiple routes are available)

This section describes these router activities and explains how you can influence the router's routing decisions under different circumstances.

Local and Wide Area Network Configuration

An XNS network is configured on each port where XNS packets are received and sent. Figure 231 is an example showing a wide area router connecting two local Ethernet networks (Santa Clara) to two wide area networks (Los Angeles and Santa Barbara).

Figure 231 Wide Area Router Connecting Four XNS Networks



Any physically attached network, Ethernet or serial line, is considered a directly connected network or "local" network. If more than one serial line is assigned to one port, that port is considered a single directly connected XNS network.

A router must check its routing table to determine where to route a packet. If the destination is on an attached network, the router can send it directly to the network. But if the destination is not directly connected, the router must route the packet to another router (called a *gateway*) that is closer to the destination. The route to a remote network can be statically configured or dynamically learned through routing protocols, such as the Routing Information Protocol (RIP) for XNS.

Defining Routes

The following sections describe the two types of routes (static and dynamic) and how to define them.

Static Routes

A static route is a user-defined route by which a remote network can be reached. To define a static route, enter the `ADD -IDP ROUTe` command and specify the appropriate route information. For more information on setting the `ROUTE` parameter, see the IDP Service Parameters chapter in *Reference for Enterprise OS Software*.

For example, on router 1 in Figure 231, you can add a static route for the Los Angeles network as follows:

```
ADD -IDP ROUTe &1122 &2233%08000100abcc
```

To display the table of static routes, enter:

```
SHow -IDP ROUTe
```

Once a static route is configured for a specific destination network, no dynamic routes will be added for that destination network.

You must configure the router with a network number (see “Displaying Routing Information” later in this chapter) before the router will accept static routes.

Dynamic Routes

Dynamic routes are routes that are learned dynamically through RIP. RIP allows the periodic exchange of routing table information with other XNS routers. Gateways use this information to route packets to other networks. For more information on this protocol, see “Learning Routes” later in this chapter.

Enhancing the Performance of the XNS Router

This section describes ways that you can enhance the performance of the XNS router.

Configuring for RIP Updates

You can change the way the router broadcasts update packets using parameters in the RIPXNS Service (see Table 56).

Table 56 Configuring the XNS Router for RIP Updates Using RIPXNS Parameters

| Parameter | Result |
|-----------------------------------|---|
| UpdateTime | Changes the frequency at which the router sends update packets. |
| CONTROL parameter options: | |
| Enabled Disabled | Determines whether router sends update packets. |

Table 56 Configuring the XNS Router for RIP Updates Using RIPXNS Parameters

| Parameter | Result |
|-------------------------|--|
| Trigger NoTrigger | Determines whether a route change for a network triggers an update packet from the router. |
| Poison NoPoison | Determines how router handles entries learned from another router. |
| OldNbrMap NewNbrMap | Permits neighbor address mapping for any bridge/router software versions. If your software version is earlier than 5.0, use option OldNbrMap. If your version is 5.0 or later, use option NewNbrMap (this is the default). |
| GlobBcast NoGlobBcast | Determines whether XNS global broadcast packets are forwarded to all interfaces except the incoming port. |

The RIPXNS parameters are automatically configured to their default values when you configure the -IDP CONTROL parameter for routing. In some cases, however, you may want to change the default configuration.

To modify the RIPXNS parameters, see the following parameter descriptions:

- CONTROL

The -RIPXNS CONTROL parameter determines on a per-port basis how the router sends the routing table information to the network. The following are the default values for the RIPXNS parameters:

CONTROL = (Enabled, Trigger, NoPoison, NewNbrMap, GlobBcast)

The impact of setting the -RIPXNS CONTROL parameter to Enabled depends on the setting of the -IDP CONTROL parameter. Table 57 shows the relationship of the -IDP CONTROL parameter to the -RIPXNS CONTROL parameter.

Table 57 CONTROL Parameters in IDP and RIPXNS

| CONTROL Setting in IDP | CONTROL Setting in RIPXNS | Effect |
|------------------------|---------------------------|---|
| Route | Enabled | Packet routing starts. Enables routing table updates based on packets received from other gateways. Routing table update packets are generated and sent to other networks. Allows normal routing performance. |
| NoRoute | Enabled | Packet routing stops. Allows routing table updates based on the packets received. Routing table update packets are not generated and sent to other networks. Allows normal routing performance when packet routing resumes. |
| Route | Disabled | Packet routing starts. Packets are routed according to static routes only. Routing table updates received are ignored. Routing table updates are not generated and sent to other networks. Reduces the amount of network data traffic and allows network administrator control over packet routing. |
| NoRoute | Disabled | Packet routing stops. Routing table updates stop (no packets are received or generated). |

Setting the `-RIPXNS CONTROL` parameter to `Trigger` causes the router to send an update packet when the network topology changes. The advantage is that triggered updates immediately allow the network to know a potentially better route to a particular network. Setting the `-RIPXNS CONTROL` parameter to `NoTrigger` reduces the amount of data packets broadcast over the network, and normal update packets are sent only at the time interval specified by the `UpdateTime` parameter.

Setting the `-RIPXNS CONTROL` parameter to `Poison` causes the router to set the number of hops for a specific table entry to 16 when it sends routing table updates. It does this to prevent routing loops in which two gateways are trying continually to update each other with the same information. The poisoned information (specified by a hop count of 16) remains in the router's update packet, adding to the data traffic on the network.

Setting the `-RIPXNS CONTROL` parameter to `NoPoison` prevents the router from sending poisoned routing information in an update packet, thus reducing the amount of data traffic over the network.

- `UpdateTime`

The `-RIPXNS UpdateTime` parameter specifies the time interval by which the router sends its routing table updates. For networks that seldom experience topology changes, the interval time can be set higher than the default value to reduce the amount of network traffic. For networks that often experience topology changes, this value can be set lower than the default value.



The lower you set the `UpdateTime` value, the more data traffic is generated on the network. Increased traffic can degrade network performance.

Configuring for Error Checking

In addition to routing configuration changes available through the RIPXNS Service parameters, you can configure the `-IDP CONTROL` parameter to provide error checking through the `Checksum | NoChecksum` value. Checksum provides a high degree of reliability in detecting bad data sent over the network. If Checksum is enabled, a router verifies the IDP checksum of a packet before it forwards the packet. The cost of this service, however, is lower network performance. The default value is `NoChecksum`.

To configure the router to provide error checking, enter:

```
SETDefault -IDP CONTROL = Checksum
```

How the XNS Router Works

This section provides general information about XNS routing.

Learning Routes

Normally, every 30 seconds (by default) or every time it learns a route change for a network, the router uses broadcast packets to report to its neighbors the following types of information:

- The networks it can reach
- The number of hops associated with each network it can reach

You can configure some router parameters to determine how the router sends out the updates by completing the following tasks:

- Changing the frequency of broadcast traffic.
- Configuring the router so that it does not send or receive update and request packets.
- Configuring the router not to send out a trigger update response when it learns a route change for a network.

Displaying Routing Information

The routing table provides information that determines how a packet is routed. The long form of the routing table displays only the most efficient route.

To display the long form, enter:

```
SHow -IDP AllRoutes Long
```

The following is a typical example of the long form of the routing table:

```
-----XNS Routing Table-----
Port      NETnumber      Gateway      Hops
1         &00003145*     &00003140%080002015980  2
1         &00003147     &00003140%080002015982  5
1         &00003149*     &00003140%080002015980  7
Total route(s) displayed: 3
```

Asterisks in the display indicate static routes.

Depending on the `AllRoutes` option selected, the routing table can include the following information, which determines how a packet is routed:

- Port number

This is the port associated with the attached network.

- Network number

The router maintains valid routes to remote networks. A network route is used to reach all hosts on the network. If you have a large routing table, you can specify a network number to verify its reachability by using the `SHow -IDP AllRoutes <NETnumber>` syntax.

- Gateway address

This is the XNS address of the gateway to which a router must send the packet before the packet can be routed to the destination. For more information on gateway addresses, see “Static Routes” earlier in this chapter.

- Number of hops between router and destination

The numbers of hops is equal to the number of gateways traversed. The XNS router selects the most efficient path for information. The most efficient path is the path that requires the fewest hops to reach a destination. In cases where two paths require the same number of hops, the router selects the first entry in the routing table.

For each destination address, the router can support up to two routes (that is, two gateways). These routes, either learned or configured, are stored in the routing table. The router selects the most efficient route to reach a destination. For information on how the router makes routing decisions, see “Learning Routes” earlier in this chapter.

To display the short form of the routing table, enter:

SHow -IDP AllRoutes

The short form, which is the default, only displays network numbers and hop counts.

Deleting Routes

Routes in the routing table are deleted differently depending on whether they are static or dynamic routes:

- A static route can be removed using the `DElete -IDP ROUte` command.

For example, to delete the Ethernet static route configured in “Static Routes” earlier in this chapter, enter:

DElete -IDP ROUte &1122

- A dynamic route learned through RIP is deleted when the router's internal timer (approximately three times the value of the `-RIPXNS UpdateTime` parameter) expires.

For example, if the `UpdateTime` parameter is set to 30 seconds, the route is deleted if no RIP updates are received for the route within 90 seconds.

To remove all dynamic routes, enter:

FLush -IDP AllRoutes

Network Reachability and Split Horizon

The types of networks that are considered *reachable* when a router broadcasts its RIP update packets are as follows:

- All directly connected networks
- All static routes
- Dynamic routes learned through RIP and currently in the routing table

Some networks, though accessible, are not reported by the router. For example, in Figure 232, router B broadcasts an update packet on network &2222. The packet does not include network &1111, because this network is learned from the same port on which the packet is broadcast. This process is known as *split horizon*.

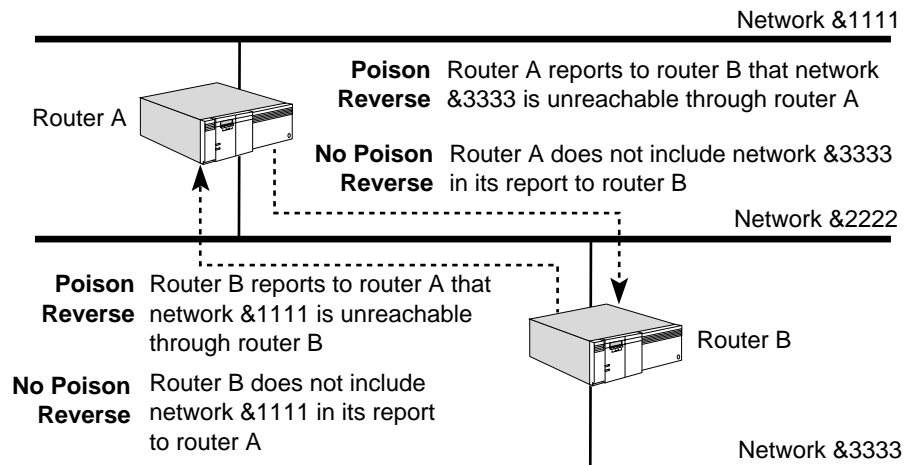
Split horizon prevents routing loops caused by including routes in the updates sent to the port from which the routes were originally learned.

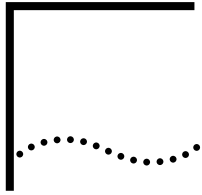
When no poison reverse is used, the router omits this type of route from routing updates sent to the same port.

With poison reverse, the router includes this type of route in its report, but the number of hops associated with that network is 16. For example, with poison reverse, router A includes networks &1111 and &3333 in its report sent to router B, but specifies that the number of hops for network &3333 is 16, while the number of hops for network &1111 is 1. Because RIP considers any network with a hop number higher than 15 unreachable, router B, upon receipt of the report, knows that packets destined for network &3333 should never be routed to router A. Through this same process, router A will know network &1111 is unreachable through router B.

Split horizon guarantees that if router B's connection with network &3333 fails, it will not send packets to router A, under the assumption that router A can reach the destination network (&3333), because it cannot.

Figure 232 XNS Routing Using Split Horizon





CONFIGURING THE LLC2 DATA LINK INTERFACE

This chapter describes the steps for configuring the Logical Link Control, type 2 (LLC2) data link interface. You may need to configure the LLC2 data link interface if you are configuring source route bridging, Advanced Peer-to-Peer Networking (APPN) routing, data link switching (DLSw), or Synchronous Data Link Control (SDLC).

Configuring LLC2 Data Link Interface

Logical Link Control, type 2 (LLC2) is a connection-oriented version of the LLC data-link layer protocol used to connect end devices. The LLC2 data link interface can be configured on the bridge/router. These parameters determine the session interaction between the LLC2 end systems and the bridge/router. The default settings should be sufficient for most network configurations.

For more information on LLC2 p-bits (poll bits) and I-frames (information frames) configured in this procedure, see the *IBM Token-Ring Network Architecture Reference* document.

To configure the LLC2 data link interface, follow these steps:

- 1 Configure the length of time the bridge/router waits for a response of an LLC2 p-bit command, or acknowledgment of an LLC2 I-frame using:

```
SETDefault !<port> -LLC2 TImeRReply = <milliseconds>(5000-60000)
```

The default is 3000 milliseconds.

- 2 Configure the length of time that the bridge/router will wait before acknowledging the received I-frame using:

```
SETDefault !<port> -LLC2 TImeRAck = <milliseconds>(0-500)
```

The default is zero.

- 3 Configure the time period that the bridge/router expects to receive a frame from the other end using:

```
SETDefault !<port> -LLC2 TImeRIInact = <milliseconds>(3000-180000)
```

The default is 60,000 milliseconds.

The bridge/router transmits a poll and activates the Reply Timer (configured in step 1) after the specified expiration time.



The TImeRIInact value should be at least five times the value entered for the TImeRReply parameter.

- 4 Define the retry count, or the maximum number of times to retransmit after the reply timer expires using:

```
SETDefault !<port -LLC2 ReTryCount = <retrys>(1-255)
```

The default is 7.

- 5 Configure the maximum frame size in bytes of the information field using:

```
SETDefault !<port> -LLC2 MaxFrame = <size>(128-4399)
```

The default is 1500 bytes.

- 6 Configure the receive window size for I-frames using:

```
SETDefault !<port> -LLC2 ReceiveWindow = <size>(1-128)
```

The default is 1.

- 7 Configure the transmit window size for I-frames using:

```
SETDefault !<port> -LLC2 TransmitWindow = <size>(1-128)
```

The default is 7.

- 8 Prepare for the number of LLC2 sessions you plan to have using:

```
SETDefault -SYS CONNecTionUsage = [High | Medium | Low]
```

This command sets up the number of LLC2 and X.25 connection service sessions allowed at one time. The default for this parameter is "High" for systems using the Dual Processor Engine (DPE) and "Low" for all other systems. If the CONNecTionUsage is set to Low, the setting may not be enough depending on how many LLC2 sessions you plan to have.

After setting this parameter, you must reboot the system for it to take effect.

You may need to change the setting to medium for some situations. For example, if you have a few LLC2 sessions running on the low setting, and you cannot get additional LLC2 sessions to connect, you may want to change the setting to medium. In addition, if you are accepting many incoming LLC2 tunnel sessions to a bridge/router serving an systems network architecture (SNA) host, you may want to change the setting to medium.

The number of LLC2 sessions possible with each CONNecTionUsage setting depends on several factors, including:

- The hardware platform and the amount of available memory
- The number of X.25 connection service sessions being run

For more information on the CONNecTionUsage parameter, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*.

Displaying LLC2 Information

You can display information about specific LLC2 sessions or a log of LLC2 activity.

To display information regarding LLC2 data link interface sessions, enter:

```
SHoW -LLC2 SESSions
```

The display is similar to the following:

```
-----LLC2 Sessions-----
.....LLC2 Active Source Mac Address:%02608C3C36AC.....
Source:%02608C3C36AC Sap:04 Dest:%02608C1A0CE7 Sap:04 Port:!2-ACTIVE
RIF: 06F0 (SRF LF=0x38:92&3:258
```

In this display, "Source" refers to the media access control (MAC) address of the bridge/router where the LLC2 connection originated, and "Destination" refers to the MAC address of the bridge/router where the LLC2 connection is intended to go. For tunneling, the source address is the peer MAC address, and the destination address is the local MAC address that is configured with the TUNnelMAcadd parameter.

The source or destination depends on which tunnel peer bridge/router you are using. For example, if you entered the SHoW -LLC2 SESSions command on the

destination bridge/router shown in the preceding display, the MAC addresses would be reversed, as shown in the following display:

```
-----LLC2 Sessions-----
.....LLC2 Active Source Mac Address:%02608C1A0CE7.....
Source:%02608C1A0CE7 Sap:04 Dest:%02608C3C36AC Sap:04 Port:12-ACTIVE
RIF: Transparent Frame
```

For more information on the parameters in the LLC2 Service, see the LLC2 Service Parameters chapter in *Reference for Enterprise OS Software*.

You can display a log of LLC2 activity by entering:

```
SHoW -LLC2 LlC2LOG
```

The log displays a history of the most recent 256 log entries including the following actions:

- Session activation or deactivation
- Session failure

Configuring LLC2 with Other Services

IBM-related services such as DLSw and APPN are affected by parameter settings in the BRidge, SR, and LLC2 Services. NETBuilder token ring ports that send or receive LLC2 or NetBIOS packets must be configured properly to avoid token ring frame copy errors and to allow connectivity. Table 58 shows the required settings in source route (SR), source route transparent (SRT), and transparent bridging environments for each of the IBM-related services. 3Com recommends configuring token ring ports for source route only mode if possible.

In Table 58, tunneling refers to the 3Com proprietary method of LLC2 tunneling, DLSw refers to data link switching, and LNM refers to LAN Net Manager. The settings are shown in abbreviated form. 3Com-recommended configurations are shown in bold.

Table 58 IBM-Related Settings for Token Ring Ports

| Services | Port Configuration | Source Route Bridging (-SR SRB) | Transparent Bridging (-BR TB) | Bridging (-BR CONT) | Route Discovery (-SR RD) | LLC2 CONTROL (-LLC2 CONT) | Frame Copy Errors |
|----------------------------|--------------------|---------------------------------|-------------------------------|---------------------|--------------------------|---------------------------|-------------------|
| Bridging only | SR | SRB | NTB | BR | NoLLC2 | Disable | None |
| Bridging only | SRT | SRB | TB | BR | NoLLC2 | Disable | * |
| Bridging only | T | NSRB | TB | BR | NoLLC2 | Disable | * |
| LNM | SR | SRB | NTB | BR | LLC2 | Enable | None |
| DLSw/
Tunneling | SR | SRB | NTB | NBR BR | LLC2 | Enable | None |
| DLSw/ Tunneling | SRT | SRB | TB | BR | LLC2 | Enable | * † |
| DLSw/ Tunneling | T | NSRB | TB | BR | NoLLC2 | Enable | * † |
| APPN | SR | SRB | NTB | NBR BR | LLC2 | Disable | None |
| APPN | SRT | SRB | TB | NBR BR | LLC2 | Disable | * |
| APPN | T | NSRB | TB | NBR BR | LLC2 | Disable | * |
| Default Setting | SRT | SRB | TB | NBR | NoLLC2 | Disable | None |

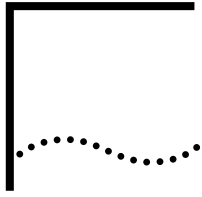
* In this configuration, end systems may generate a small number of token-ring MAC frame copy error report packets when the NETBuilder bridge/router is initializing or when it ages out a MAC address from its bridge table.

† In this configuration it is important for global bridging to be enabled, otherwise the token ring hardware does not filter transparent packets. This can generate many frame copy error reports and adversely effect performance. To prevent forwarding of bridge packets in this configuration, enter the following command: SETDefault -BRidge CONTrol = NoForward. The NoForward parameter allows DLSw and LLC2 tunneling to send and receive LLC2 SNA and NETBios packets, but prevents other packets from bridging.

The row in Table 58 labeled DLSw/Tunneling with port configuration SR represents DLSw or 3Com tunneling in a source-route-only port configuration. The entries in this row expand to the following NETBuilder software configuration syntax:

```
SETDefault -BRidge CONTrol = Bridge | NoBridge
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
SETDefault !<port> -BRidge TransparentBridge = NoTransparentBridge
SETDefault !<port> -SR RingNumber = <number> (1-4095)
SETDefault !<port> -SR RouteDiscovery = LLC2
SETDefault !<port> -LLC2 CONTrol = Enable
```

In this configuration, global bridging is enabled or disabled on one or more token ring ports. Transparent bridging is disabled, source routing and route discovery are configured, and LLC2 is enabled.



CONFIGURING SNA NETWORKS USING QLLC TO LLC2 CONVERSION

This chapter describes how to configure a NETBuilder bridge/router to use the Qualified Logical-Link Control (QLLC) to LLC2 conversion to allow an SNA network to transfer data link control information between adjacent SNA nodes over an X.25 network.



For conceptual information, see “How QLLC to LLC2 Conversion Works” later in this chapter.

Setting Up QLLC to LLC2 Conversion

This section describes how to configure your bridge/router to transmit and receive data over an X.25 interface using QLLC to LLC2 conversion.

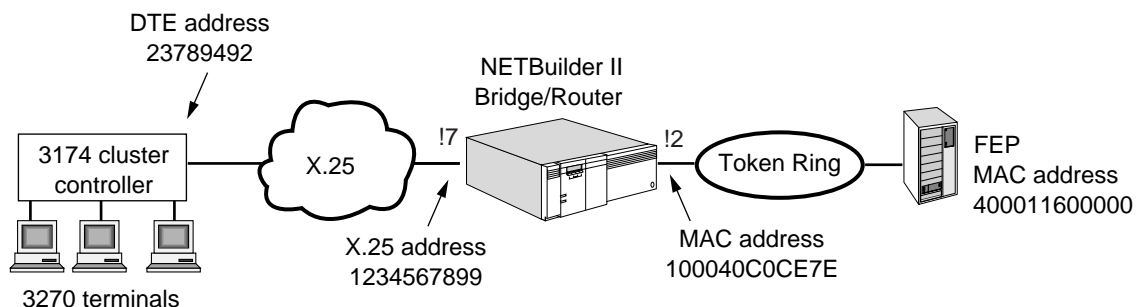
For detailed descriptions of all commands, see *Reference for Enterprise OS Software*.

Prerequisites Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your bridge/router ports, virtual ports, and paths according to the Configuring Basic Ports and Paths chapter.
- If using DLSw, configure your bridge/router’s DLSw service according to the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

Procedure To set up QLLC to LLC2 conversion to transmit SNA packets over an X.25 network, see Figure 233 and follow these steps:

Figure 233 Setting up QLLC



1 Establish X.25 as the port owner by entering:

```
SETDefault !7 -PORT OWNEr=X25
```



You may need to perform additional X.25 interface configuration according to the needs of your installation. See the *Configuring Wide Area Networking Using X.25* chapter for information about configuring X.25.

2 Add a QLLC CU on port 7 using:

```
ADD !<port> -QLLC PortCU <CU Name> <CU DTE Addr (1-15 digits)> <Local MAC>
    [<Remote MAC>] [Local SAP] [Remote SAP]
```

For example enter:

```
ADD !7 -QLLC PORTCU QLLCCU1 23789492 100040C0CE7E 400011600000
```

3 If switched virtual circuits are to be used on the QLLC connection, configure SVCs in the X.25 component by entering:

```
SETDefault !QLLCCU1 -QLLC CUVCTYPE=SVC
SETDefault !7 -X25 twowaySVCs=11,20
SETDefault !7 -X25 X25address=1234567899
SETDefault !7 -PA CONT=ENabled
```

Enabling the path is required for the X.25 parameters to take effect.

4 If permanent virtual circuits (PVCs) are to be used on the QLLC connection, configure the PVCs in the X.25 component by entering:

```
SETDefault !QLLCCU1 -QLLC CUVCTYPE=PVC
SETDefault !7 -X25 PVC 1,1 23789492 c3 0
SETDefault !7 -PA CONTROL=ENable
```

Enabling the path is required for the X.25 parameters to take effect.

5 Enable the QLLC CU and the port by entering:

```
SETDefault !QLLCCU1 -QLLC CUCONT=Enable
SETDefault !7 -QLLC PCONT=ENable
```

6 Enable LLC2 on the LAN port by entering:

```
SETDefault !2 -LLC2 CONT=ENable
```

How QLLC to LLC2 Conversion Works

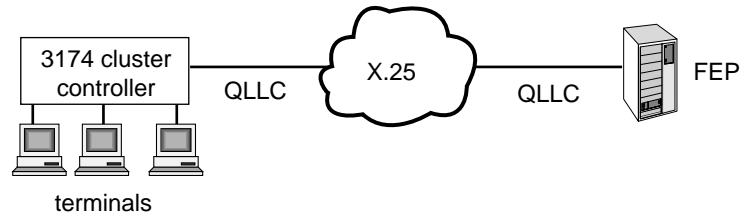
Qualified Logical-Link Control (QLLC) is a sub-layer defined by IBM, which is a link-level control protocol positioned between the X.25 Packet Layer and SNA's Path Control. This sub-layer provides services that allow SNA to transfer data link control information between adjacent SNA nodes over an X.25 network by using X.25 data packets.

In the traditional SNA over X.25 environment, the NCP Packet Switched Interface (NPSI) software is used on the Front-end Processors (FEPs). NPSI is a costly and CPU intensive interface which is difficult to configure and maintain.

Converting QLLC to LLC2 frames on the NETBuilder bridge/router allows NPSI to be removed from the FEPs. The converted LLC2 frames are transported across WAN links to the data center either to a NETBuilder bridge/router and to the FEP via a LAN interface or, in the case of RFC 1490 transport, directly to a FEP over a Frame Relay connection. The converted LLC2 frames can also be bridged into the FEP with either TIC or Ethernet interfaces.

To transport SNA over the X.25 interface, the SNA-based units must attach to a non-SNA Packet Switched Data Network (PSDN) as an X.25 DTE. A typical SNA host environment, see Figure 234, uses the NPSI to support attachment of a PSDN to FEP:

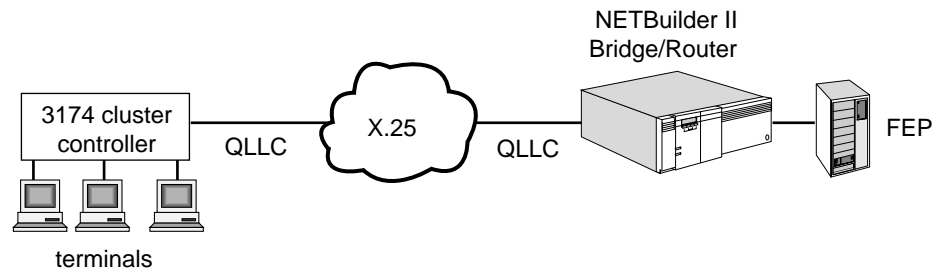
Figure 234 Typical Configuration Using NPSI



The NPSI encapsulates and decapsulates SNA message units with the packet headers, so the message conforms to the X.25 interface and can be transported over the PSDN.

To replace the NPSI, the NETBuilder bridge/router plays the NPSI role. In addition, the QLLC packet, which is encapsulating an SNA message with X.25 packet and link headers, is converted to LLC2 by the NETBuilder bridge/router. See Figure 235.

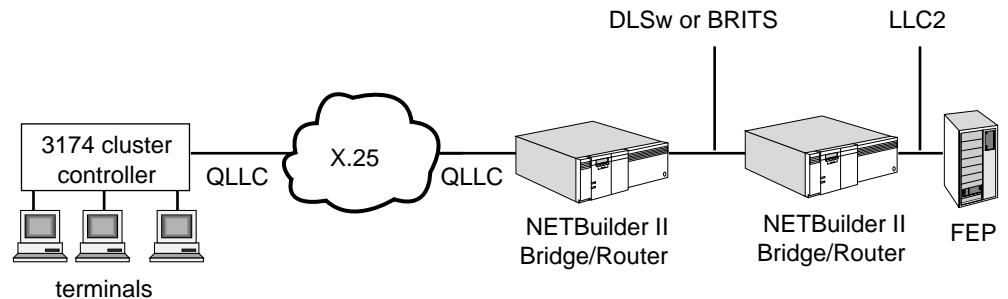
Figure 235 Using QLLC Conversion on a NETBuilder



The LLC2 packet is switched either locally or through an IP network via DLSw to another NETBuilder bridge/router before it reaches the FEP. After the replacement of NPSI with NETBuilder bridge/routers, the FEP is no longer configured with an X.25 interface. The FEP can then use the NCP Token Ring interface or any other LAN interface to attach to the NETBuilder bridge/router directly.

QLLC to LLC2 conversion is also available with DLSw, RFC 1490, or BRITS (Boundary Routing for SNA) as the WAN transport mechanism. See Figure 236.

Figure 236 Using QLLC Conversion on a NETBuilder with DLSw or BRITS



QLLC Acronyms

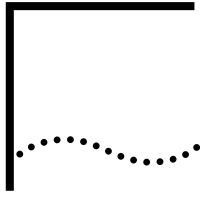
The following acronyms are used in this chapter to explain QLLC:

| | |
|------|--|
| FEP | Front End Processor |
| NCP | Network Control Program, a software application running on a FEP. |
| NPSI | NCP Packet Switching Interface, a software application running on a FEP. |
| CC | Cluster Controller, a typical IBM 3x74 device. |
| CU | Control Unit which represents a remote DTE device. |
| QLLC | Qualified Logical Link Control, a protocol also known as LLC3. |
| QLM | QLLC Link Manager, used to convert QLLC to/from LLC2. |
| PSDN | Packet Switched Data Network |

Limitations

In this software release only Primary QLLC functions are implemented. Secondary or Peer functions are not supported. The SNA device type is limited to PU2. PU2.1 and PU1 are not supported. XID spoofing is supported for both call-in (connections initiated from the remote controllers) and call-out (connections initiated from the host.)

On the X.25 side, both PVC and SVC circuits are supported. The converted QLLC data can be transferred to FEP via DLSw, Local Switching, BAN, BNN, or BRITSS.



CONFIGURING SYNCHRONOUS DATA LINK CONTROL CONNECTIVITY

This chapter describes how to provide Synchronous Data Link Control (SDLC) connectivity over local and wide area networks, how the SDLC works on the router, and gives guidelines for operating and managing your SDLC configuration.



For conceptual information, see “How SDLC Conversion Works” later in this chapter. For information about the parameters in the SDLC Service, see the SDLC Service Parameters chapter in Reference for Enterprise OS Software.



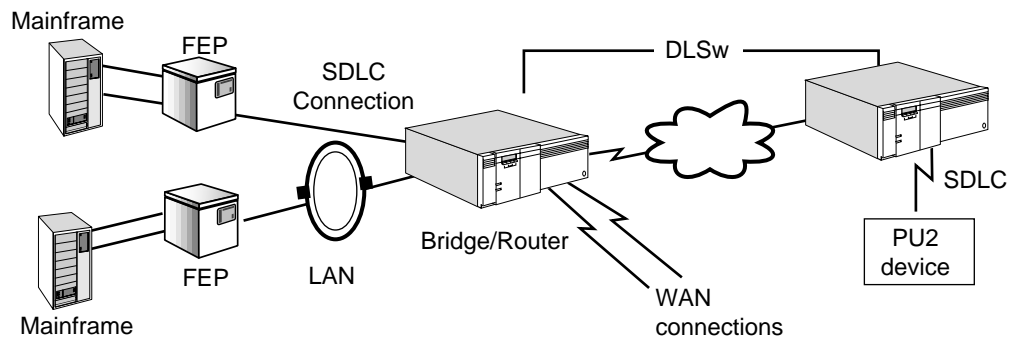
On the NETBuilder II system, SDLC is supported only on the HSS 3-Port modules.

Connection Methods

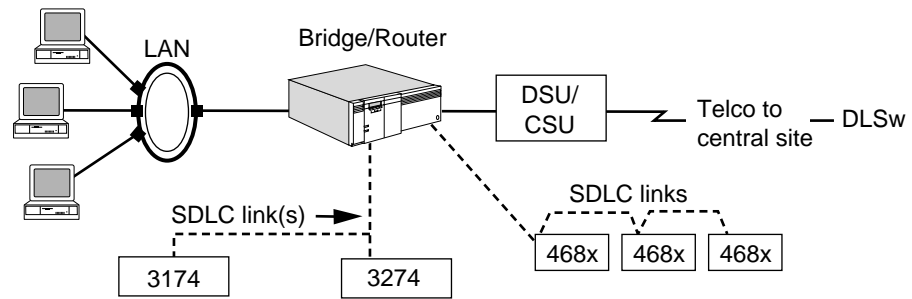
This section describes various SDLC connections. For configuration procedures, see “Configuring the Router for SDLC” later in this chapter.

Figure 237 shows an SDLC point-to-point configuration where remote PU2 devices use SDLC to connect to an SDLC- or token ring-attached host front end processor (FEP) through the WAN. In this configuration, the SNA and SDLC data is passed through the bridge/router using data link switching (DLSw).

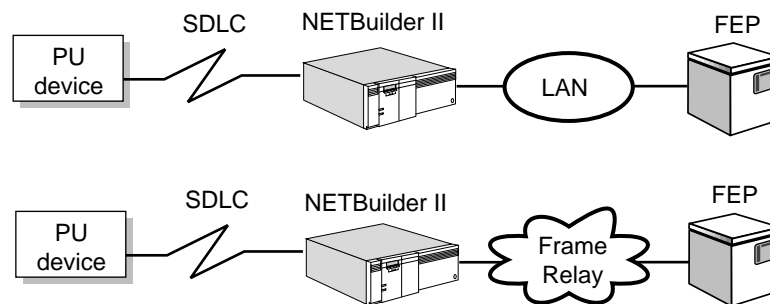
Figure 237 SDLC Point-to-Point Configuration



A multipoint configuration may consist of several remote SDLC devices using SDLC connections to a 3Com bridge/router to reduce the number of independent (SDLC and other) links required by the site. In this configuration, the SDLC data is passed through the bridge/router using DLSw. As shown in Figure 238, a remote site may be configured as an SDLC primary node talking to 3x74 cluster controllers and other SDLC secondary devices (486x).

Figure 238 SDLC Multipoint Configuration

The SDLC connectivity of the NETBuilder II bridge/router also allows an SDLC-attached device to communicate with a local LAN-attached device or with a front-end processor (FEP) through Frame Relay (see Figure 239).

Figure 239 SDLC Connectivity through a LAN and Frame Relay

Configuring the Router for SDLC

This section describes how to configure the bridge/router for SDLC. After you complete the procedures in this section, proceed to "Configuring the CU Devices on the Link" later in this chapter.

Prerequisites Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the non-SDLC ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter. For the token ring bridge see the Configuring Source Route Bridging chapter.
- Set up the ports for SDLC as described later in this guide. Ports being used for SDLC must have a one-to-one port-to-path mapping.
- Set up the LLC2 data link interface as described in the Configuring the LLC2 Data Link Interface chapter.



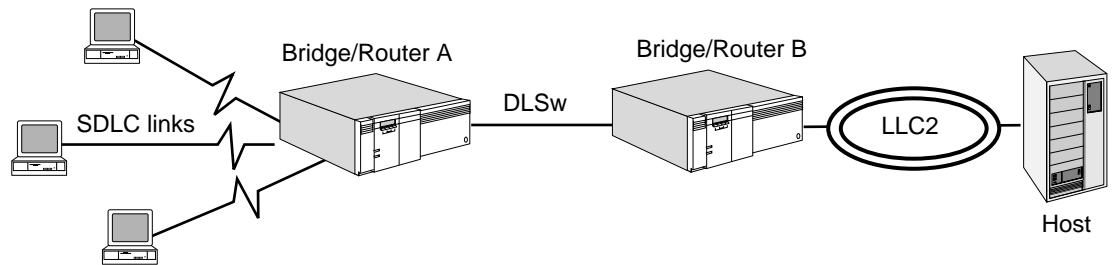
SDLC is affected by parameter settings in other services. For more information, see "Configuring LLC2 with Other Services" in the Configuring the LLC2 Data Link Interface chapter.

- Configure IP and obtain the IP addresses of the local bridge/router and DLSw peers.

- Obtain the SNA device address and the Virtual Telecommunications Access Method (VTAM) address so you can match the addresses of the devices you are configuring.
- Allocate a media access control (MAC) and Service Access Point (SAP) configuration to represent the attached SDLC device.

Procedure To configure SDLC, perform the procedures in the following section on bridge/router A (see Figure 240), which has the attached SDLC devices.

Figure 240 Configuring the Router for SDLC



Configuring the SDLC Port and Path Attributes

To configure the port attributes for SDLC, follow these steps:

- 1 For the port running SDLC, set the OWNEr parameter using:

```
SETDefault !<port> -PORT OWNEr = SDLC
```

- 2 Set the communication mode (using the PATH Service DUplex parameter) and the transmission encoding method (using the PATH Service ENCOding parameter) for the path assigned to the SDLC port.

For example, if the attached device requires half-duplex communication and nonreturn to zero (NRZ) encoding, use:

```
SETDefault !<path> -PATH DUplex = Half
SETDefault !<path> -PATH ENCOding = NRZ
SETDefault !<path> -PATH TxIdle = Mark
```

Make sure the parameter settings match the configuration of the device and that you configure the TxIdle parameter as shown. The Mark setting allows half-duplex operation to occur by setting up the system to receive the second half of the transmission without aborting.

- 3 Make sure the LineType and clocking parameters in the PATH Service are set correctly.

For a back-to-back or a null modem connection you must use external clocking. The NETBuilder II bridge/router does not provide an internal clock source. The LineType must be Leased and clocking must be External. Set the LineType and clocking parameters using:

```
SETDefault !<path> -PATH LineType = Leased
SETDefault !<path> -PATH Clock = External
```

Changes to these parameters do not take effect until you enable the ports and paths.

- 4 Disable Link Access Procedure, Balanced (LAPB) on the selected path using:

```
SETDefault !<path> -LAPB CONTrol = Disable
```

Because of the SDLC configuration, the port-to-path correlation must be mapped on a one-to-one basis. LAPB cannot be enabled when the port owner is SDLC.

Configuring LLC2 and Bridging Characteristics

If SDLC devices attached to the bridge/router need to communicate with LAN (LLC2) devices attached to the same bridge/router, the bridge/router must be set up to support LLC2 connections. You also must set up DLSw to perform internal switching. If this has not already been done, see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

If you want your NETBuilder II bridge/router to connect the SDLC devices to a LAN, the LLC2 ports must be configured as described for DLSw in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

Configuring the SDLC Protocol Characteristics

To configure SDLC for communication with the connected devices, follow these steps:

- 1 Define the control units (CUs) attached to the port.

Assign a name to the CU using this port using:

```
ADD !<port> -SDLC PortCU <CU name>
```

Assigning the CU to the port sets up the SDLC configuration of the port, which allows you to view and modify the port parameter settings. The CU name you assign has only local significance. CU names must be unique and can be no longer than 8 alphanumeric characters. A name longer than 8 characters is rejected and a warning message appears. However, maintaining name consistency between the NETBuilder II bridge/router and network control point (NCP) configurations may simplify configuration management. CU names must be unique within the bridge/router.



The SDLC parameters for a port are inaccessible. They cannot be viewed or modified until at least one CU is assigned to the port with the Port CU parameter.

- 2 Configure whether the port will act in a primary or secondary role in the connection.

In SDLC, a primary station controls the operation of other secondary stations. The role of the port applies to the port and all of the CUs configured on the port. The role must be set according to the CUs attached to the port, if the role of the CU is secondary, set the port as primary.

For example, to attach a CU device that is secondary, set the port as primary on the bridge/router using:

```
SETDefault !<port> -SDLC PROle = Primary
```

All SDLC ports that are attached to a primary (a host) should be set as secondary. All SDLC ports that are attached to a secondary CU (a 3174 downstream physical unit) should be set as primary.

- 3 Display the current parameter settings using:

```
SHow !<port> -SDLC PCONFig
```

- 4 Define whether the port operates in half- or full-duplex mode with the connected device.

To configure a port on the bridge/router for half-duplex operation, use:

```
SETDefault !<port> -SDLC PDatmode = Half
```

The port can be set for half-duplex (two-way alternating) or full-duplex (two-way simultaneous) communication to match the configuration of the SDLC devices on this port.

- 5 Set the maximum amount of data contained in a single frame (basic transmission unit (BTU) size) using:

```
SETDefault !<port> -SDLC PMaxData = 521
```

The value of this parameter should match the host.

- 6 Set the frame numbering method used by the CUs attached to this port using:

```
SETDefault !<port> -SDLC PModulo = 128
```

The setting of this parameter must match the CU on this port.

Configuring the SDLC Protocol Timing Parameters

To configure the timing of the port, which affects how the port waits and responds to communication with the CUs attached to this port, follow these steps:

- 1 Set the number of times the bridge/router attempts to complete a protocol exchange with an SDLC connected device before considering that device as having failed using:

```
SETDefault !<port> -SDLC PT1Retry = 3
```

- 2 Set the no-response time-out waiting period for the port using:

```
SETDefault !<port> -SDLC PT1Timer = 400
```

This parameter is used on primary ports only. If the CU does not send a response to a poll or a message from the SDLC port before this timer expires, the transmission is retried until the retry count set in the previous step runs out. At this point, the bridge/router will terminate (disconnect) the SDLC connection and attempt to contact the CU again for a new connection.

- 3 Set the delay between attempts to connect the network data link (LLC2) partner for CUs whose mode is set to originate using:

```
SETDefault !<port> -SDLC PRetryTimer = 20
```

After you have completed this procedure, proceed to the next section to configure the link stations (CUs) attached to the bridge/router.

Configuring the CU Devices on the Link

This section describes how to configure the SDLC connection for the CU devices the bridge/router has configured on each SDLC port.

Prerequisites

Before beginning this procedure, complete the following steps:

- Configure the SDLC port parameters as described in the previous procedure.
- Obtain the MAC/SAP address pair for the CU. This includes the local MAC/SAP values used to represent this SDLC device in the LLC2 network environment and the remote MAC/SAP values, if the CUMode parameter is set to Originate, which indicate the CUs partner (LLC2) device.

Procedure

To configure the link for the CU, follow these steps:

- 1 Define the type of CU you are configuring by entering:

```
SETDefault !LS22 -SDLC CUType = T1
```

This command specifies the CU named LS22 which is a type PU1 or T1 device.

- 2 Set the CU device identification if required using:

```
SETDefault !<CU name> -SDLC CUXId = 0179097C
SETDefault !<CU name> -SDLC CUXidDefined = Yes
```

This step is optional and depends upon the configuration of the attached CU and the requirements of the network partner device.

The CUXId is only required if the PRole parameter is set to primary and the PU type is set to type 2.0 or type 1 and the attached (secondary) CU will not respond to an exchange identification (XID) poll. The CUXidDefined parameter must also be set to enable use of the defined CUXId value.

See the SDLC Service Parameters chapter in *Reference for Enterprise OS Software* for further information about the CUXId parameter.

3 Configure the poll address of the secondary CU using:

```
SETDefault !<CU name> -SDLC CUAddr = C2
```

If the bridge/router is configured as primary, the CU address must match the PU. If the bridge/router is set as secondary, the CU address must match the host configuration.

4 Configure the local MAC address for the CU using:

```
SETDefault !<CU name> -SDLC CULocalMac = %50004080C940
```

This value is the MAC address used within the LLC2/DLSw environment to communicate with the CU. LLC2 frames intended for this CU must use this value as the destination MAC address. When the NETBuilder II bridge/router sends LLC2/DLSw frames on behalf of this CU, this value is used as the source MAC address. The MAC addresses in this parameter are in noncanonical format.



Two CUs may not use the same CULocalMac and CULocalSap combination.

5 Configure the local SAP used by the CU using:

```
SETDefault !<CU name> -SDLC CULocalSap = 08
```

This value is the LLC2 SAP used for this CU in the LLC2/DLSw environment. LLC2 frames intended for this CU must use this value as the destination SAP. When the NETBuilder II bridge/router sends LLC2/DLSw frames on behalf of this CU, this value is used as the source SAP.

6 Set the maximum number of frames that may be transmitted before waiting for a response using:

```
SETDefault !<port> -SDLC CUMaxOut = 4
```

7 Set up the operating mode for the CU by entering:

```
SETDefault !<CU name> -SDLC CUMode = Originate
```

The mode determines whether the bridge/router initiates sessions in the LLC2/DLSw environment on behalf of this CU, or whether the bridge/router simply responds to sessions initiated by other stations. The exact sequence of connection events also depends on the PRole parameter.

When CUMode is set to Originate, the bridge/router initiates an LLC2 connection as soon as it has contacted the SDLC station; that is, when it has received an XID or set normal response mode (SNRM) (if PRole is secondary) or a response to an XID or SNRM (if PRole is Primary).

When CUMode is set to Answer, the bridge/router will not initiate LLC2 sessions, but responds to LLC2 connection attempts by trying to establish contact with the

SDLC station, sending XID or SNRM if PRole is Primary, and responding to XID or SNRM if PRole is secondary.

- 8 Configure the remote MAC address for the CU using:

```
SETDefault !<CU name> -SDLC CUREmoteMac = %60003070C940
```

When CUMode is Originate, the bridge/router uses this value as the destination MAC address when initiating an LLC2 connection on behalf of this CU.

- 9 Configure the remote SAP used by the CU using:

```
SETDefault !<CU name> -SDLC CUREmoteSap = 08
```

When CUMode is Originate, the bridge/router uses this value as the destination LSAP when initiating an LLC2 connection on behalf of this CU.

- 10 Enable SDLC for the port.

For example, to connect a CU to a bridge/router, enable the SDLC Protocol using:

```
SETDefault !<port> -SDLC PControl = Enable
```

- 11 Check that there is a one-to-one port-to-path mapping by displaying the PORT Service PATHs parameter by entering:

```
SHow !1 -PORT PATHs
```

- 12 Enable the CU using:

```
SETDefault !<CU name> -SDLC CUControl = Enabled
```

Both the port and the CU must be enabled for an SDLC link to operate. The port and the CU may be enabled in any order. Be sure you have enabled the CONTROL parameter in the PORT Service and PATH Service.

Verifying the Configuration

After you have configured SDLC, you can display SDLC port and CU configuration information. You can also add or delete CUs assigned to a port. Deleting all CUs assigned to a port also deletes the SDLC configuration of the port.

To display all of the SDLC port parameters configured for the specified port and the CU configuration for all CUs assigned to that port, enter:

```
SHow -SDLC PCONFig
```

To display the SDLC configuration for a specific port, for example port 1, enter:

```
SHow !1 -SDLC PCONFig
```

Display the value of all CU-related parameters for each CU by entering:

```
SHow !* -SDLC CUCONFig
```

To display a specific CU, use:

```
SHow !<CU name> -SDLC CUCONFig
```

You can display a log of SDLC activity by entering:

```
SHow -SDLC sdlcLOG
```

The log displays a history of the most recent 256 log entries including the following actions:

- Control unit activation or deactivation

- Control unit failure

Using Frame Relay Access

If SDLC devices attached to the bridge/router need to communicate with a FEP attached directly to the bridge/router through Frame Relay, the bridge/router must be set up to support the mapping of LLC2 traffic to Frame Relay. If you have not already configured Frame Relay, perform the procedures in the *Configuring Wide Area Networking Using Frame Relay* chapter.

APPN over SDLC

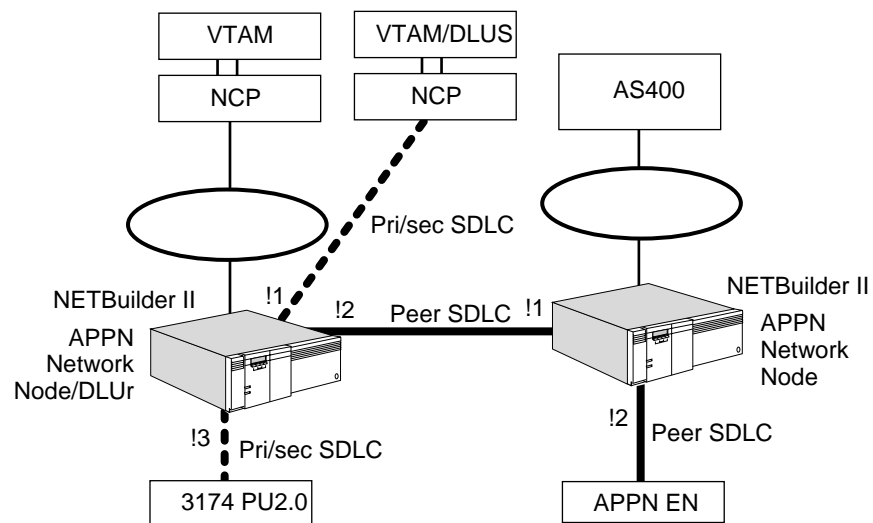
You can configure Advanced Peer-to-Peer Networking (APPN) traffic to run over SDLC. To configure the bridge/router network node to run over SDLC, you first configure the SDLC port and path attributes between the network node and the partner node using the procedures in this chapter. You then configure the APPN network node following the procedures in the *Configuring APPN Intermediate Session Routing* chapter; the procedures are similar to configuring APPN over other data link control (DLC) types, except that you do the following tasks:

- When setting the -APPN PortDef parameter for APPN ports, set the DLC type to SDLC and optionally, set the DatMode and ROle values.
- Configure adjacent link stations using the -APPN SdlcAdjLinkSta parameter and configure Dependent LU Requestor (DLUr) link stations using the -APPN SdlcDlurLinkSta parameter.

For information about these APPN Service parameters, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

Figure 241 shows a configuration in which APPN traffic is being sent over SDLC connections. The NETBuilder II bridge/router acting as a network node is shown sending APPN over SDLC connections to a peer NETBuilder II network node, to a network control program (NCP), and to a 3174 PU2.0 type node. The network node is also serving as a DLUr for the VTAM Dependent LU server (DLUs).

Figure 241 APPN Traffic over SDLC



You can send either APPN ISR traffic or APPN HPR traffic over SDLC connections. For information about configuring APPN Intermediate Session Routing (ISR) over

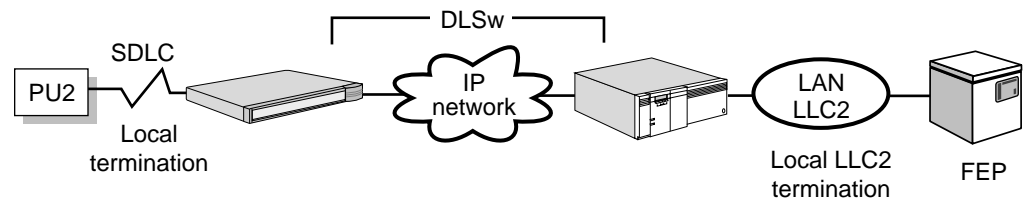
SDLC, follow the procedures in the Configuring APPN Intermediate Session Routing chapter. For information about configuring APPN High Performance Routing (HPR), see the APPN High Performance Routing chapter.

How SDLC Conversion Works

SDLC devices generally are referred to as physical units (PUs), control units (CUs), and linkstations. CU is used in parameter descriptions and names. The term PU is used in some examples and general discussion and when referring to device type. The term link station may also be used to see the device type. Except for parameter names or specific device types, these terms may be used interchangeably.

SDLC connectivity allows SDLC devices to communicate with local or remote non-SDLC (LLC2, Frame Relay devices, or other remote SDLC devices, using an SDLC connection to your bridge/router. SDLC polling and response occur locally between the SDLC device and the NETBuilder II bridge/router; the SNA data stream is tunneled through the network using the DLSw protocol. Figure 242 shows a typical SDLC/DLSw configuration.

Figure 242 SDLC/DLSw Configuration



The NETBuilder II bridge/router provides SDLC connectivity by mapping the SDLC device to a "virtual" LLC2 device, or an LLC2 device to an SDLC device. Other systems in the network communicate with this LLC2 device, for example, through DLSw. The bridge/router passes the data to and from the SDLC device.

In addition to the configuration information required to communicate with the SDLC device, the bridge/router must be given the LLC2 information used to communicate with non-SDLC devices, for example, MAC and SAP values.

Operating the bridge/router with SDLC is accomplished by mapping (or conversion) of SDLC connections into LLC2 connections. This section describes the key aspects of this mapping, how the configuration parameters relate to (and affect) the mapping behavior, and illustrates the configuration for common applications.

Both SDLC and LLC2 are reliable data link protocols. They provide sequenced, acknowledged, and retransmitted delivery of data frames, and include special frames for session initiation and termination. Although similar, SDLC and LLC2 operate in different environments using different modes. Because of this different frame sequences are used for session initiation, and different addressing schemes are used in each service.

Data Link Switch is a protocol (defined by RFC 1434) used to link LLC2 sessions together across an internetworked reliable transport protocol. The protocol

definition for DLSw includes addressing schemes and session startup sequences that readily correspond to LLC2 addressing and session startup.

To link SDLC sessions with LLC2 or DLSw sessions, the bridge/router must handle two major functions: address mapping and session initiation.

Address Mapping

In an LLC2 (LAN) or DLSw environment, addressing consists of MAC/SAP pairs. Each station has a MAC (LAN) address; every frame sent from one station (A) to another station (B) contains both a source and a destination MAC address to distinguish the sending and receiving stations on the shared-access medium. Each LLC2 frame also contains a source and destination LLC2 SAP (LSAP). A pair of LAN stations may have multiple sessions between them; the LSAP values are used to distinguish between frames belonging to different sessions.

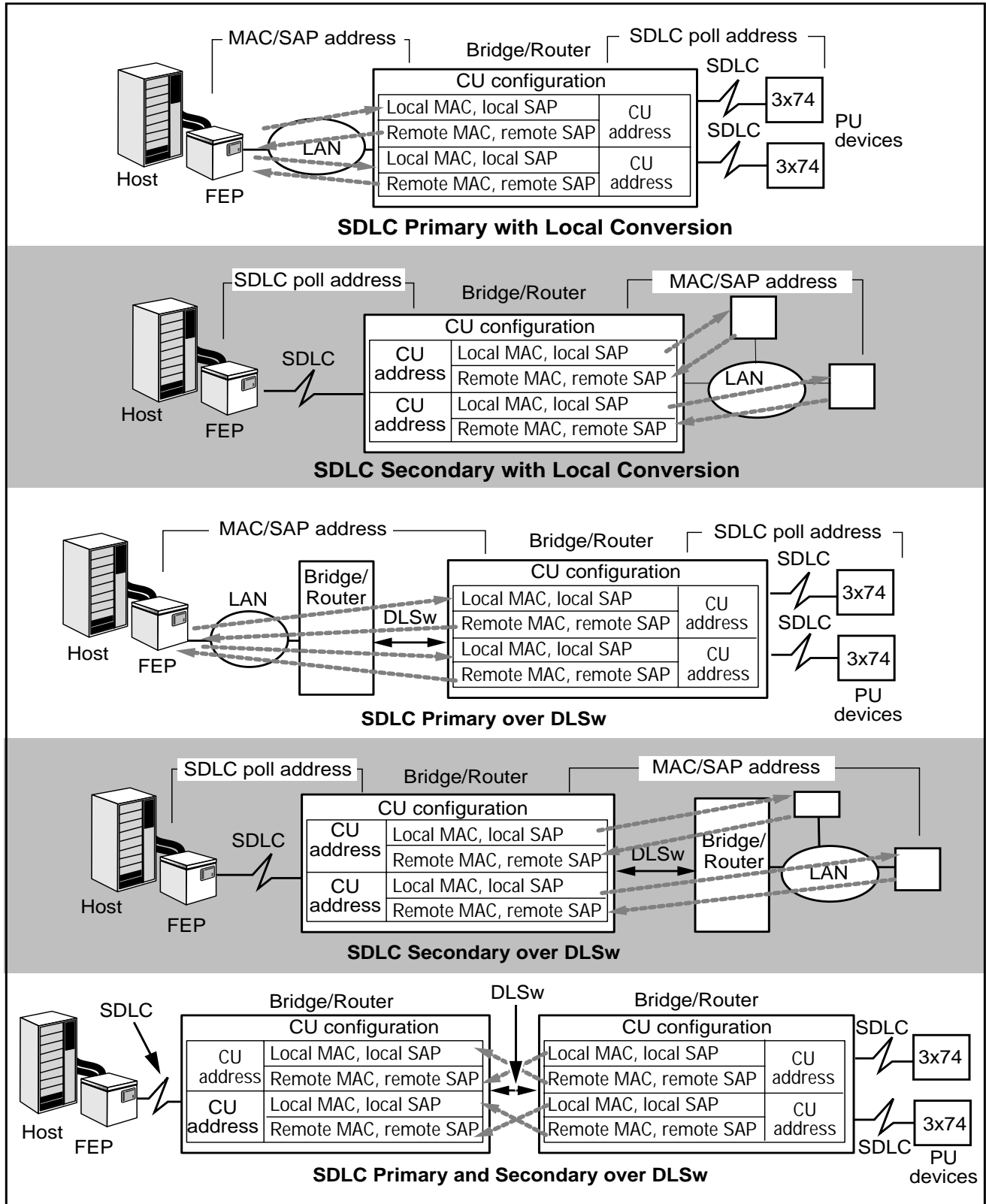
In an SDLC environment, each frame carries only a single address value; the identifier of the secondary station that is to receive, or that sent, this frame.

To allow stations in an LLC2 environment to communicate with an attached SDLC station, the bridge/router maps a set of LLC2 (LAN) addresses to each CU. This address mapping is configured using the CULocalMac, CUREmoteMac, CULocalSap, and CUREmoteSap parameters in the SDLC Service. The bridge/router appears as a LAN-based CU mapped to the SDLC CU, which uses the CULocalMac and CULocalSap to communicate with other LAN stations. Figure 243 shows various types of address mapping for SDLC.

In the LLC2 or DLSw environment, the CUs supported by the bridge/router appear to be attached to the LAN: Frames can be sent to them using the MAC/SAP values assigned as CULocalMac and CULocalSap. Frames sent by the bridge/router on behalf of the CU use the CULocalMac and CULocalSap values as the source address values.

If the bridge/router is initiating an LLC2 session on behalf of a CU (see "Session Initiation" next), it must know which LLC2 station to send the connection request to. This destination is determined by the CUREmoteMac and CUREmoteSap parameter values. These values are used as the destination for LLC2 frames when the bridge/router initiates such sessions.

Figure 243 SDLC Address Mapping



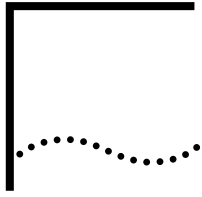
Session Initiation

In addition to address mapping, the SDLC configuration determines how the bridge/router initiates SDLC and LLC2/DLSw connections. The CUMode parameter for the specific CU, in conjunction with the SDLC role of the port (the PROle parameter) determines the system's behavior as follows:

- In an SDLC environment, datalink connections are initiated by the bridge/router acting as the primary station. The primary station controls the operation of other secondary stations. When acting as an SDLC secondary station, the bridge/router accepts connections only. Use the CUMode parameter to configure the bridge/router to either originate a network connection request or answer a connection request from the network.
- In an LLC2 or DLSw environment, either of the two systems involved in a session may initiate the datalink connection. The CUMode parameter also determines whether and when the NETBuilder II bridge/router initiates LLC2 or DLSw connections.

When CUMode is set to Originate, the bridge/router initiates datalink connections in the LLC2/DLSw environment. Contact is achieved by sending an XID or SNRM and receiving a response. When the bridge/router is secondary, contact is achieved by receiving an XID or SNRM from the primary station. However, if the bridge/router is a secondary station in Originate mode, it will not respond until it has completed the LLC2/DLSw connection.

When the CUMode is set to Answer, the bridge/router will only accept datalink connections from the LLC2/DLSw environment. When a connection request is received, the bridge/router attempts to set up the corresponding SDLC connection before accepting the LLC2/DLSw connection. When you set up the SDLC connection, an XID or SNRM is sent if the bridge/router is acting as a primary station; if the bridge/router is acting as a secondary station, an XID or SNRM response is made.



CONFIGURING SDLC AND HDLC TUNNELING FOR SNA NETWORKS

This chapter describes how to configure tunneling for synchronous data link control (SDLC) and high-level data link control (HDLC) traffic using the IBM Data Link Switching (DLSw) protocol.



For conceptual information on how SDLC and HDLC tunneling works, see “How SDLC and HDLC Tunneling Works” later in this chapter. For information about the SHDlc Service parameter, see the SHDlc Service Parameters chapter in Reference for Enterprise OS Software.

Configuring SDLC and HDLC Tunneling

This section describes how to configure the bridge/router for SDLC and HDLC tunneling, by referring to Figure 244. The figure shows an IBM host connected to an IBM controller through NETBuilder bridge/routers and an IP network.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the IP addressing and IP routing protocols on the appropriate ports according to the Configuring IP Routing chapter.
- Obtain the IP addresses for bridge/routers on each side of the TCP/IP connection.

Procedure

Configure ports, paths, SDLC and HDLC tunneling, and data link switching by referring to the example in Figure 244 and completing the steps under “Configuring Router A” and “Configuring Router B.” Table 59 lists the commands used in these steps.

Figure 244 SDLC and HDLC Tunneling Example

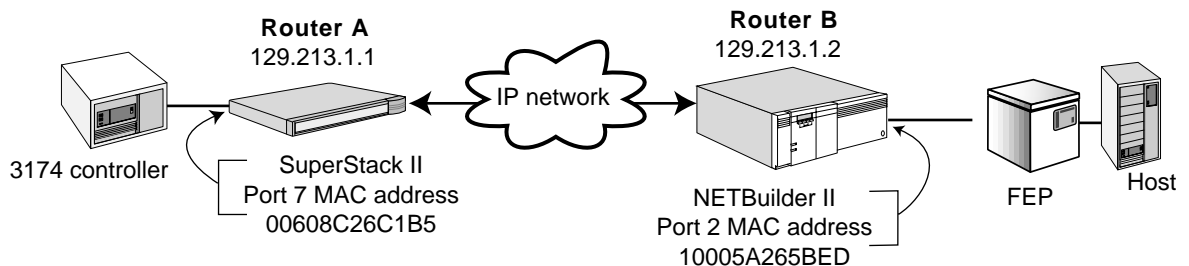


Table 59 Commands to Configure SDLC and HDLC Tunneling and Data Link Switching

| Commands Entered on Bridge/Router A | Commands Entered on Bridge/Router B |
|--|--|
| SHow -PORT CONFIguration | SHow -PORT CONFIguration |
| SETDefault !1 -IP NETaddr = 129.213.1.1 | SETDefault !1 -IP NETaddr = 129.213.1.2 |
| SETDefault !7 -PORT OWNer = SHDlc | SETDefault !2 -PORT OWNer = SHDlc |
| SETDefault !7 -PATH CLock = External | SETDefault !2 -PATH CLock = External |
| SETDefault !7 -PATH CONNector = RS232 | SETDefault !2 -PATH CONNector = RS232 |
| SETDefault !7 -PATH LineType = Leased | SETDefault !2 -PATH LineType = Leased |
| SETDefault !7 -PATH BAud = 19.2 | SETDefault !2 -PATH BAud = 19.2 |
| SETDefault !7 -PATH DUplex = Full | SETDefault !2 -PATH DUplex = Full |
| SETDefault -TCP CONTRol = KeepALive | SETDefault -TCP CONTRol = KeepALive |
| SETDefault -TCP KeepALive = 3 | SETDefault -TCP KeepALive = 3 |
| SETDefault -DLSw Interface = 129.213.1.1 | SETDefault -DLSw Interface = 129.213.1.2 |
| SETDefault -DLSw CONTRol = (EnableSNA, DisableNetBIOS) | SETDefault -DLSw CONTRol = (EnableSNA, DisableNetBIOS) |
| SETDefault !7 -PATH CONTRol = Enable | SETDefault !2 -PATH CONTRol = Enable |
| ADD !1 -DLSw PEer 129.213.1.2 | ADD !1 -DLSw PEer 129.213.1.1 |
| SystemInfo | SHow -SYS MacAddrFormat |
| | SETDefault !2 -SYS MacAddrFormat = Noncanonical |
| SHow -PATH MacAddress | SHow -PATH MacAddress |
| SETDefault !7 -SHDlc PEer = %10005A265BED | SETDefault !2 -SHDlc PEer = %00608C26C1B5 |

Configuring Router A

To configure router A, follow these steps:

- 1 Display the port configuration by entering:

```
SHow -PORT CONFIguration
```

The display shows the ownership status of each port.

- 2 Define the IP address for the port through which the router is going to tunnel by entering:

```
SETDefault !1 -IP NETaddr = 129.213.1.1
```

- 3 Set the port ownership of serial port 7 to SHDlc by entering:

```
SETDefault !7 -PORT OWNer = SHDlc
```

When you use a WAN port, you need to configure the port owner. SDLC and HDLC tunneling use only WAN ports.



The number of SHDlc ports a NETBuilder II bridge/router can support is the number of WAN paths it can operate simultaneously.

- 4 Display attributes for all available paths by entering:

```
SHow -PATH CONFIguration
```

- 5 Set the attributes for the SHDlc line by entering:

```
SETDefault !7 -PATH CLock = External
```

```
SETDefault !7 -PATH CONNector = RS232
SETDefault !7 -PATH LineType = Leased
SETDefault !7 -PATH BAUD = 19.2
SETDefault !7 -PATH DUplex = Full
```

If you use a single- port WAN adapter, set the -PATH ENCOding parameter to NRZ.



SHDLC only supports full-duplex operation.

After configuring values using the PATH Service, you may receive a message telling you to re-enable the path. If you receive this message, re-enable the path with the SETDefault !<path> -PATH CONTrol = Enable command.

- 6 Enable the transmission of TCP keepalive packets by entering:

```
SETDefault -TCP CONTrol = KeepAlive
```

TCP keepalive packets notify the bridge/router when the TCP connection has ended. Without TCP keepalive packets, the bridge/router will not detect that the TCP connection is down due to an abnormal situation. The data link switching sessions may remain active even though the corresponding TCP session has ended.

- 7 Specify the number of contiguously missed keepalive packets that brings down the TCP session.

For example, if you want three retries, enter:

```
SETDefault -TCP KeepAliveLimit = 3
```

- 8 Configure an IP address to connect traffic to and from the router.

The address must be one that has been defined in the router using the SETDefault !<port> -IP NETaddr syntax. This address is the only address used for data link switching.

If you are configuring your NETBuilder II bridge/router as an IP router, the port associated with this IP address must be active before any packets can be sent to or received by this IP address. Select an IP address associated with a port that is always up or is the most reliable, such as a LAN port.

To map the specified DLSw tunnel to the local IP address of bridge/router A, enter:

```
SETDefault -DLSw Interface = 129.213.1.1
```



All Internet addresses for connected bridge/routers must be known in the local bridge/router's routing table, either dynamically through RIP or OSPF, or statically configured in the IP routing tables.

- 9 Enable data link switching for SNA traffic on the port by entering:

```
SETDefault -DLSw CONTrol = (EnableSNA, DisableNetBIOS)
```

This setting allows SNA traffic to flow through the data link switch and disables NetBIOS traffic. SNA traffic must be enabled for SDLC and HDLC tunneling to work.

If you are going to use the prioritization feature of DLSw, see "Prioritizing DLSw Traffic" in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter before proceeding to the next step.

- 10 Configure the DLSw tunnel peer IP connection by entering the following command. You also must specify the tunnel ID, a peer network address, and optionally, a name for the tunnel connection.

```
ADD !1 -DLsw PEer 129.213.1.2
```

When configuring the Internet address for the tunnel peer, you do not specify the port number of the bridge/router where the connection will be made. When a tunnel connection is made, the bridge/router determines the port through which the peer Internet address can be reached. When a peer has been defined and enabled, the system continuously retries to connect to the peer until a TCP connection is established between the system and the peer.

- 11** From the peer router, display the format of the peer MAC address by entering:

```
SHow -SYS MacAddrFormat
```

- 12** To display the local MAC address on the SuperStack bridge/router, enter:

```
SystemInfo
```

- 13** To convert the MAC address to noncanonical format, you must enter the MacAddrConvert command on a NETBuilder II bridge/router. This command is not available on the SuperStack bridge/router.

- 14** Display the peer router MAC address by entering:

```
SHow -PATH MacAddress
```

With this display, you can obtain the peer router MAC address that you configure in the next step.

- 15** Set the MAC address for the peer serial port that the local SDLC port is communicating with by entering:

```
SETDefault !7 -SHDlc PEer = %10005A265BED
```

Configuring Router B

To configure router B, repeat steps 1–11 in the preceding procedure, then continue with the following steps (performed on a NETBuilder II bridge/router):

- 1** If the peer MAC address displayed is in canonical format, set it to noncanonical by entering:

```
SETDefault !7 -SYS MacAddrFormat - Noncanonical
```

- 2** Display the peer router MAC address by entering:

```
SHow -PATH MacAddress
```

With this display, you can obtain the peer router MAC address that you configure in the next step.

- 3** Set the MAC address for the peer serial port that the local SDLC port is communicating with using:

```
SETDefault !<port> -SHDlc PEer
```

Verifying the Configuration

After you have configured a tunnel connection using data link switching, you can display information to verify the connection.

To display complete configuration information, enter:

```
SHow -DLsw CONFIguration
```

The display shows the settings you have configured.

To display the peer information, enter:

```
SHow -SHDlc -PEer
```


The following display is an example of this information:

```
-----SHDlc PEer-----
Local Port   Local MacAddress   Circuit State Peer MacAddress
      4           %1000405011DC     CONNECTED   %100040605D8A
```

When shown in the display, SAP E8 represents an HDLC tunnel.

Displaying Circuits

To display the status of circuits, enter:

```
SHow -DLsw CIRcuits
```

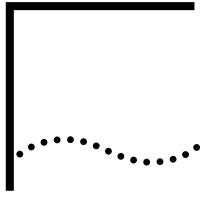
Information similar to the following is displayed:

```
-----Circuits-----
Local          DL Corr.  Port Peer          DL Corr.  State Peer IP
Name/Address   Name/Address
%00608C26C1B5  04 376B008E  4   %10005A265BED  04 37250047  CONNECTD 129.213.1.2
%0020AF00DCC8  04 171A00C9  6   %40000003172A  04 86FA0013  CONNECTD 200.200.1.254
%100040600B03  00 00000000  ?   %100040A0E8E1  00 00000000  DISC      200.200.1.254
%400001111111  08 00000000  ?   %400002222222  08 00000000  DISC      200.200.1.254
%400011600000  04 00000000  ?   %0020AF00B940  04 00000000  DISC      200.200.1.254
%400011600000  34 00000000  ?   %0020AFEE9630  34 00000000  DISC      200.200.1.254
```

How SDLC and HDLC Tunneling Works

The SDLC and HDLC tunneling features enable NETBuilder II bridge/routers to send SDLC or HDLC frames across IP networks through DLSw tunnels. Two bridge/routers interconnect a point-to-point SDLC or HDLC link. They encapsulate SDLC or HDLC frames sent between the two end points and tunnel them through an IP network.

A typical use of SDLC and HDLC tunneling is to connect a host computer and a remote terminal or controller. In Figure 244, two end points of an SDLC link (a 3174 controller and a host) are interconnected by two intermediary bridge/routers. The 3174 controller is connected to port 7 on router A, and the host FEP is connected to port 2 on router B.



CONFIGURING DATA LINK SWITCHING FOR SNA AND NETBIOS NETWORKS

This chapter describes how to configure data link switching on your system to connect networks running IBM Systems Network Architecture (SNA) and NetBIOS traffic over Transmission Control Protocol/ Internet Protocol (TCP/IP) using the IBM Data Link Switching (DLSw) Protocol.



For conceptual information on how data link switching works, see “How Data Link Switching Works” later in this chapter and RFC 1795. The 3Com implementation of DLSw is based on this standard. Also, to simplify configuration, you can use DLSw multicast. For more information, see the Configuring Multicast Data Link Switching for NetBIOS and SNA Networks chapter.

Configuring for SNA

This section describes how to configure the bridge/router for both ends of an SNA connection using the DLSw protocol.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the token ring LAN as described in the Configuring Source Route Bridging chapter.
- Configure the IP addressing and IP routing protocols on the appropriate ports according to the Abbreviations and Acronyms chapter.
- Obtain the IP addresses for bridge/routers on each side of the TCP/IP connection.

Figure 245 shows a sample data link switching configuration for an SNA environment.

Figure 245 Configuring Data Link Switching for SNA

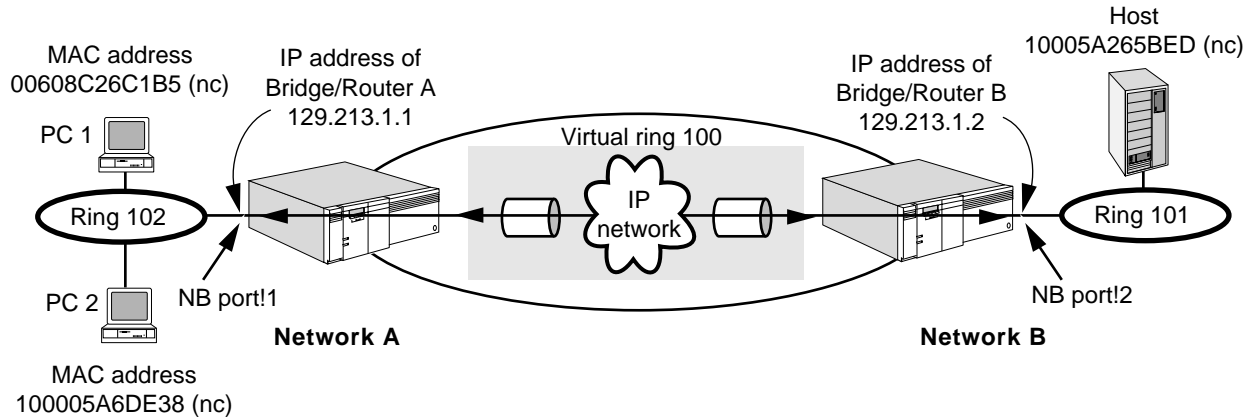


Table 60 lists the commands used to configure the example in Figure 245.

Table 60 Commands to Configure Data Link Switching for SNA

| Commands Entered on Bridge/Router A | Commands Entered on Bridge/Router B |
|---|---|
| SETDefault -TCP CONTROL = KeepAlive | SETDefault -TCP CONTROL = KeepAlive |
| SETDefault -TCP KeepAlive = 3 | SETDefault -TCP KeepAlive = 3 |
| SETDefault !1 -LLC2 CONTROL = Enable | SETDefault !2 -LLC2 CONTROL = Enable |
| SETDefault !1 -SR RouteDiscovery = LLC2 | SETDefault !2 -SR RouteDiscovery = LLC2 |
| SETDefault -LLC2 TUNnelVRing = 100 | SETDefault -LLC2 TUNnelVRing = 100 |
| SETDefault -DLSw MODE = Secure | SETDefault -DLSw MODE = Secure |
| SETDefault -DLSw Interface = 129.213.1.1 | SETDefault -DLSw Interface = 129.213.1.2 |
| ADD !1 -DLSw PEer 129.213.1.2 | ADD !1 -DLSw PEer 129.213.1.1 |
| SETDefault -DLSw CONTROL = EnableSNA,
DisableNetBios | SETDefault -DLSw CONTROL = EnableSNA,
DisableNetBios |

Procedure

To configure data link switching for SNA bridge/router A, see Figure 245 and follow these steps:

- 1 Enable transmission of Transmission Control Protocol (TCP) keepalive packets by entering:

```
SETDefault -TCP CONTROL = KeepAlive
```

TCP keepalive packets notify the bridge/router when the TCP connection has ended. Without TCP keepalive packets, the bridge/router will not detect that the TCP connection is down due to an abnormal situation. This can result in data link switching sessions being kept active even though the corresponding TCP session has ended.

- 2 Specify the number of contiguously missed keepalive packets that will bring down the TCP session.

For example, if you want three retries, enter:

```
SETDefault -TCP KeepAliveLimit = 3
```

- 3 Enable LLC traffic from a port to be tunneled through data link switching by entering:

```
SETDefault !1 -LLC2 CONTROL = Enable
```

This command enables LLC2 traffic on port 1.

Enable route discovery by entering:

```
SETDefault !1 -SR RouteDiscovery = LLC2
```

This command enables route discovery for LLC2 on port 1.

Repeat this step for each port you are configuring.



DLSw is affected by parameter settings in other services. For more information, see "Configuring LLC2 with Other Services" in the Configuring the LLC2 Data Link Interface chapter.

- 4 Assign a unique virtual ring number for the data link switching cloud.

This ring number is used by source routing and some data link switching Switch-to-Switch Protocol (SSP) messages. For example, to configure the virtual tunnel ring, enter:

```
SETDefault -LLC2 TUNnelVRing = 100
```

When using source routing, the internetwork becomes a virtual ring. If your end systems are using token ring source routing, the bridge/router and the IP tunnel appear to the end systems as a source route bridge with a token ring network attached to the other side of the bridge/router.



This virtual ring number must match on all peer bridge/routers used for tunneling and must be unique within the token ring network.

- 5 Configure the desired mode of operation.

To configure secure mode, enter:

```
SETDefault -DLSw MMode = Secure
```

The router accepts connections only from data link switches defined in the ADD PEer parameter.

To configure for default prioritization, enter:

```
SETDefault -DLSw MMode = Secure, DefaultPRioritized
```

You can also configure the mode to multicast. For more information, see the Configuring Multicast Data Link Switching for NetBIOS and SNA Networks chapter.

- 6 Configure an IP address to connect traffic to and from the router.

The address must be one that has been defined in the router using the SETDefault !<port> -IP NETAddr syntax. This address is the only address used for data link switching.

If you are configuring your NETBuilder II bridge/router as an IP router, the port associated with this IP address must be active before any packets can be sent to or received by this IP address. Select an IP address associated with a port that is always up or is the most reliable, such as a LAN port.

To map the specified DLSw tunnel to the local IP address of bridge/router A, enter:

```
SETDefault -DLSw Interface = 129.213.1.1
```



All Internet addresses for connected bridge/routers must be known in the routing table of the local bridge/router, either dynamically through RIP or OSPF, or statically configured in the IP routing tables.

- 7 Enable data link switching for SNA traffic on the port by entering:

```
SETDefault -DLsw CONTROL = (EnableSNA, DisableNetBIOS)
```

This setting allows SNA traffic to flow through the data link switch and disables NetBIOS traffic.

If you are going to use the prioritization feature of DLSw, see “Prioritizing DLSw Traffic” later in this chapter before proceeding to step 8.

- 8 Configure the DLSw tunnel peer IP connection by entering:

```
ADD !1 -DLsw PEer 129.213.1.2
```

When configuring the Internet address for the tunnel peer, you do not specify the port number of the bridge/router where the connection will be made. When a tunnel connection is made, the bridge/router determines the port through which the peer Internet address can be reached. When a peer has been defined and enabled, the system continuously retries to connect to the peer until a TCP connection is established between the system and the peer.

- 9 To configure bridge/router B, repeat steps 1–8.

Configuring for NetBIOS

This section describes how to configure data link switching for NetBIOS traffic.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the token ring LAN as described in the Configuring Source Route Bridging chapter.
- Configure the IP addressing and IP routing protocols on the appropriate ports according to the Configuring IP Routing chapter.
- Obtain the IP addresses for both bridge/routers on either side of the TCP/IP connection.

Procedure Figure 246 shows a sample DLSw configuration for a NetBIOS environment.

Figure 246 Configuring Data Link Switching for NetBIOS

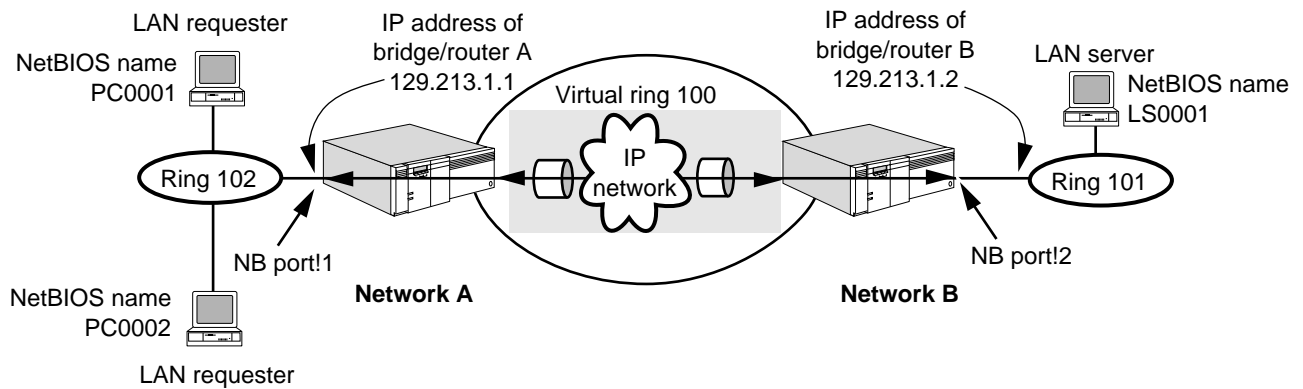


Table 61 lists the commands used for this configuration.

Table 61 Commands to Configure Data Link Switching for NetBIOS

| Commands Entered on Bridge/Router A | Commands Entered on Bridge/Router B |
|--|--|
| SETDefault -TCP CONTROL = KeepALive | SETDefault -TCP CONTROL = KeepALive |
| SETDefault -TCP KeepALive = 3 | SETDefault -TCP KeepALive = 3 |
| SETDefault !1 -LLC2 CONTROL = Enable | SETDefault !2 -LLC2 CONTROL = Enable |
| SETDefault !1 -SR RouteDiscovery = LLC2 | SETDefault !2 -SR RouteDiscovery = LLC2 |
| SETDefault !1 -SR RingNumber = 102 | SETDefault !2 -SR RingNumber = 101 |
| SETDefault -LLC2 TUNnelVRing = 100 | SETDefault -LLC2 TUNnelVRing = 100 |
| SETDefault -DLSw Interface = 129.213.1.1 | SETDefault -DLSw Interface = 129.213.1.2 |
| ADD !1 -DLSw PEer 129.213.1.2 | ADD !1 -DLSw PEer 129.213.1.1 |
| SETDefault -DLSw CONTROL = DisableSNA, EnableNetBios | SETDefault -DLSw CONTROL = DisableSNA, EnableNetBios |

To configure data link switching for NetBIOS on bridge/router A, see Figure 246 and follow these steps:

- 1 Enable transmission of TCP keepalive packets by entering:

```
SETDefault -TCP CONTROL = KeepALive
```

TCP keepalive packets notify the bridge/router when the TCP connection has ended. Without TCP keepalive packets, the bridge/router will not detect that the TCP connection is down due to an abnormal situation. The data link switching sessions may be kept active even though the corresponding TCP session has ended.

- 2 Specify the number of contiguously missed keepalive packets that will bring down the TCP session.

For example, if you want three retries, enter:

```
SETDefault -TCP KeepAliveLimit = 3
```

- 3 Enable LLC traffic from a port to be tunneled through data link switching by entering:

```
SETDefault !1 -LLC2 CONTROL = Enable
```

This command enables traffic on port 1.

Enable route discovery by entering:

```
SETDefault !1 -SR RouteDiscovery = LLC2
```

This command enables route discovery for LLC2 on port 1.

Assign the ring number to the local port by entering:

```
SETDefault !1 -SR RingNumber = 102
```

Repeat this step for each port you are configuring.

- 4 Assign a unique virtual ring number for the data link switching cloud.

This ring number is used by source routing and some data link switching SSP messages. For example, to configure the virtual tunnel ring, enter:

```
SETDefault -LLC2 TUNnelVRing = 100
```

When using source routing, the internetwork becomes a virtual ring. If your end systems are using token ring source routing, the bridge/router and the IP tunnel appear to the end systems as a source route bridge with a token ring network attached to the other side of the bridge/router.



This virtual ring number must match on all peer bridge/routers used for tunneling and must be unique within the token ring network.

- 5 Configure the desired mode of operation.

To configure secure mode, enter:

```
SETDefault -DLsw MOde = Secure
```

The router accepts connections only from data link switches defined in the ADD !<tunnelid> PEer parameter.

- 6 Configure an IP address to connect traffic to and from the router.

The address must be one that has been defined in the router using the SETDefault !<port> -IP NETaddr syntax. This address is the only address used for data link switching. To map the specified DLSw tunnel to the local IP address of bridge/router A, enter:

```
SETDefault -DLsw Interface = 129.213.1.1
```



All Internet addresses for connected bridge/routers must be known in the routing table of the local bridge/router, either dynamically through RIP or OSPF, or statically configured in the IP routing tables.

- 7 Enable data link switching for NetBIOS traffic on the port by entering:

```
SETDefault -DLsw CONTROL = (EnableNetBios, DisablesNA)
```

This setting allows NetBIOS traffic and disables SNA traffic from flowing through the data link switch.

If you are going to use the prioritization feature of DLSw, see "Prioritizing DLSw Traffic" *later in this chapter* before proceeding to step 8.

- 8 Configure the DLSw tunnel peer IP connection by entering:

```
ADD !1 -DLsw PEer 129.213.1.2
```

When configuring the Internet address for the tunnel peer, you do not specify the port number of the bridge/router where the connection will be made. When a

tunnel connection is made, the bridge/router determines the port through which the peer Internet address can be reached.

- 9 To configure bridge/router B, repeat steps 1–8, and enter the addresses of the session partners.

Verifying the Configuration

After you have configured a tunnel connection using data link switching, you can display information to verify the connection.

To display complete configuration information, enter:

```
SHoW -DLsw CONFIguration
```

The display shows the settings you have configured.

Displaying Connections

To display the control status of your connections between two data link switched SNA networks, enter:

```
SHoW -DLsw CONNectiOns
```

The display shows whether data link switching has established a connection with the Peer IP address. For example, the following display shows the results of the sample configuration in Figure 245 on the bridge/router A side:

```
-----Connections-----
Peer IP Address          State                No. of Circuits
129.213.1.2             ACTIVE              0
```

To determine whether TCP has established connections, display TCP port information, and to verify whether the two data link switching port numbers are active or connected, enter:

```
SHoW -TCP CONNectiOns
```

This display shows the actual TCP connections. There are two connections for each DLSw tunnel.

```
-----TCP Connection Table-----
Loc Address      Port    Rem address      Port    State    ConnID
129.213.1.1     2065   129.213.1.2     2067   estab   1966085
129.213.1.1     2067   129.213.1.2     2065   estab   1900550
```

Displaying Circuits

To display the status of circuits, enter:

```
SHoW -DLsw CIRcuits
```

Information similar to the following is displayed:

```
-----Circuits-----
Local          DL Corr.  Port Peer          DL Corr.  State  Peer IP
Name/Address   Name/Address
%00608C26C1B5  04 376B008E  4  %10005A265BED  04 37250047  CONNECTD  129.213.1.2
%0020AF00DCC8  04 171A00C9  6  %40000003172A  04 86FA0013  CONNECTD  200.200.1.254
%100040600B03  00 00000000  ?  %100040A0E8E1  00 00000000  DISC      200.200.1.254
%400001111111  08 00000000  ?  %400002222222  08 00000000  DISC      200.200.1.254
```

```
%40001160000 04 00000000 ? %0020AF00B940 04 00000000 DISC 200.200.1.254
%40001160000 34 00000000 ? %0020AFEE9630 34 00000000 DISC 200.200.1.254
```

For more information about the possible states, see the CIRcuits parameter in the DLSw Service Parameters chapter of *Reference for Enterprise OS Software*.

Displaying LLC Sessions

Logical link control (LLC) displays media access control (MAC) addresses in canonical format. Use the MacAddrConvert command to convert a MAC address in canonical format to noncanonical format. To display the status of configured sessions, enter:

```
SHow -LLC2 SESSions
```

Information similar to the following is displayed:

```
-----LLC2 Sessions-----
.....LLC2 Active Source Mac Address:%0020AF1D2C10
Source:%0020AF1D2C10 Sap:04 Dest:%02608C1A0CE7Sap:04 Port:11 -ACTIVE
RIF: Transparent Frame
```

Displaying Cache

You can also display the contents of the names in your NetBIOS names cache by entering:

```
SHow -DLsw NameCache
```

The cache displays both static and dynamic names:

```
-----Netbios Names Cache-----
Peer: IP Address           Netbios Name
129.213.1.2               LANSERVER1
```

You can display the contents in the MAC addresses cache by entering:

```
SHow -DLsw MacCache
```

When verifying MAC addresses for the sample configuration shown in Figure 245, information similar to the following is displayed:

```
-----Mac Addresses Cache-----
Peer: IP Address           Mac Address
129.213.1.2               %100051265BED
```

Displaying the DLSw Activity Log

You can display a log of DLSw activity by entering:

```
SHow -DLsw DLswLOG
```

The log displays a history of the most recent log entries including the following actions:

- Circuit activation or deactivation
- Circuit failure
- Tunnel activation or deactivation
- Tunnel failure
- Capabilities exchange accepted or rejected

The following display is an example of this log:

```
50 Tue May 28 17:51:54 1996 Capex Ack IP 192.100.2.3      Vectors: 81 82 83 86 84
#49 Tue May 28 17:51:54 1996 Capex Ack IP 192.100.100.1  Vectors: 81 82 83 86 84
#48 Tue May 28 17:51:54 1996 Tunnel UP IP 192.100.2.3
#47 Tue May 28 17:51:54 1996 Tunnel UP IP 192.100.100.1
#46 Tue May 28 17:09:11 1996 Circuit DOWN LMAC 100040501175 LSAP 0C RMAC 100040 5011DB RSAP 0C IP
192.100.2.3
#45 Tue May 28 17:09:11 1996 Circuit DOWN LMAC 100040501175 LSAP 08 RMAC 100040 5011DB RSAP 08 IP
192.100.2.3
#44 Tue May 28 17:09:11 1996 Circuit DOWN LMAC 100040501175 LSAP 04 RMAC 100040 5011DB RSAP 04 IP
192.100.2.3
```

Displaying the DLSw End-Station Topology

You can configure the bridge/router to collect end-station topology information for the DLSw network topology and display it to help troubleshoot the network. Before you can display the topology, you must first specify whether you want logical unit (LU) topology or physical unit (PU) topology information collected.

To collect end-station topology use:

```
SETDefault -DLSw SnaTopoCollect = (EnablePu | EnablePuLu | Disable)
```

To collect end-station topology PU and LU information, enter:

```
SETDefault -DLSw SnaTopoCollect = EnablePuLu
```

To collect end-station topology PU information only, enter:

```
SETDefault -DLSw SnaTopoCollect = EnablePU
```

To disable the collection of either LU or PU information, enter:

```
SETDefault -DLSw SnaTopoCollect = Disable
```

To display the DLSw topology map based on the end-station topology collection information, enter:

```
SHow -DLSw SnaTopoDisplay
```

The following display is an example showing the end-stations in a DLSw topology:

```

-----SNA End-Station Topology-----
PU Name : US3COMHQ.PU01BJ1  Node ID  : 05D 90100 Node Type: NN Dep. LU: Yes
MAC Addr: %00608C24F2F6 04  Port Num : 2          DLC Type : TR
Status  : ACTIVE           Active LU: 4
Bound LU : 1

-----
LU Name      Add      T State Pri. LU  | LU Name  Add T State Pri.LU
-----+-----
LU01BJ1      2        2 BOUND CNM01LU  |           3        ACTIVE
4            ACTIVE                    |           5        ACTIVE
=====
PU Name :                Node ID  : 017 9079D  Node Type: 2.0 Dep. LU: Yes
MAC Addr: %10004060532C 04  Port Num : 7A      DLC Type : SDLC
Status  : ACTIVE           Active LU: 8          Bound LU : 1

-----
LU Name      Add      T State Pri. LU  | LU      Add T State Pri.LU
-----+-----
                2        ACTIVE                    |           3        PNDACT
                4        INACTIVE                   |           5        ACTIVE
                6        ACTIVE                    |           7        ACTIVE
                8        ACTIVE                    |           9        2 BOUND
                                         LUTRDSH          LU002MVS
=====
PU Name : US3COMHQ.DLSWWS1  Node ID  : 05D 00210 Node Type: EN  Dep. LU: No
MAC Addr: %00608C24F2F6 04  Port Num : 1          DLC Type : ETH
Status  :
ACTIVE
=====

```

In this display, the end-stations shown are as follows:

- The first end-station with the PU name PU01BJ1 is a token ring station with four dependent LUs in varying states. For example, the LU named LU01BJ1, is in the BOUND state with the primary LU (PLU) CNM01LU. The other LUs, which are not named, are active.



The second end-station is an SDLC 3174 with dependent LUs in varying states. It does not show a PU name because the XIDs exchanged were type 1 (and the name was not manually set).

- The third end-station, with the PU name DLSWWS1, is a PU 2.1 attached end-station through Ethernet, and has no dependent LUs.

In the display, each end-station description is separated by the double lines. This display shows both PU and LU topology information (obtained by specifying EnablePuLu for the SnaTopoCollect parameter.

If only PU information was collected, the display would not show the LU information.

For more information about this display, see the SnaTopoDisplay parameter description in the DLSw Service Parameters chapter *Reference for Enterprise OS Software*.



When an SNMP Manager such as SunNet Manager or OpenView is used, more information about each end-station is displayed than is available through the NETBuilder SnaTopoDisplay parameter display.

Customizing the Configurations

This section describes how to customize data link switching configurations.

Defining a Non-Secure Host Configuration

By default, the bridge/router can accept DLSw tunnel connections from any other configured DLSw bridge/routers. By setting a bridge/router to a Secure state, unauthorized sites can be prevented from accessing a particular site. For less vital traffic, you can leave the bridge/router configured to accept tunnel connections from any remote bridge/router. If you plan to have terminal users at many different remote sites making tunnel connections to a site, you can use the NonSecure setting.

Figure 247 shows a bridge/router at a central host site accepting incoming tunnel connections from three branch offices to access the local site.

Figure 247 Data Link Switching Tunnel Configuration for Central bridge/router

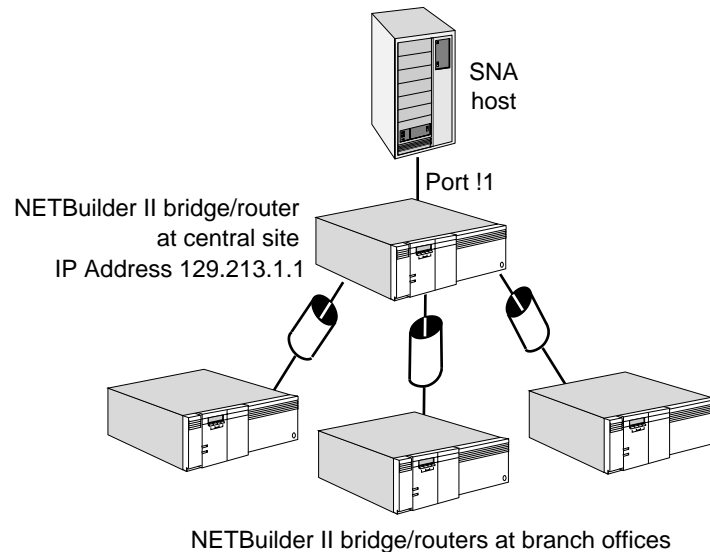


Table 62 lists the commands used for this configuration.

Table 62 Commands to Configure Data Link Switching for a Central Site Bridge/Router

Commands Entered on the Central Site Bridge/Router

```
SETDefault -TCP CONTrol = KeepALive
SETDefault -TCP KeepALive = 3
SETDefault !1 -LLC2 CONTrol = Enable
SETDefault -LLC2 TUNnelVRing = 100
SETDefault -DLSw MMode = NonSecure
SETDefault -DLSw Interface = 129.213.1.1
SETDefault -DLSw CONTrol = EnableSNA, DisableNetBios
```

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the token ring LAN as described in the Configuring Source Route Bridging chapter.
- Configure the IP addressing and IP routing protocols on the appropriate ports according to the Configuring IP Routing chapter.
- Obtain the IP addresses for both bridge/routers on either side of the TCP/IP connection.

Procedure

To configure a central site bridge/router to accept any incoming tunnel connection requests, follow these steps:

- 1 Enable transmission of TCP keepalive packets by entering:

```
SETDefault -TCP CONTROL = KeepAlive
```

TCP keepalive packets notify the bridge/router when the TCP connection has ended. Without TCP keepalive packets, the bridge/router will not detect that the TCP connection is down due to an abnormal situation. This can result in data link switching sessions being kept active even though the corresponding TCP session has ended.



This command will keep switched virtual circuits active even though there is no traffic across the link other than KeepAlive packets.

- 2 Specify the number of contiguously missed keepalive packets that will bring down the TCP session.

For example, if you want three retries, enter:

```
SETDefault -TCP KeepAliveLimit = 3
```

- 3 Enable LLC traffic from a port to be tunneled through data link switching by entering:

```
SETDefault !1 -LLC2 CONTROL = Enable
```

This command enables LLC2 traffic on port 1. Repeat this step for each port you are configuring.

- 4 Assign a unique virtual ring number for the data link switching cloud.

This ring number is used by source routing and some data link switching SSP messages. For example, to configure the virtual tunnel ring, enter:

```
SETDefault -LLC2 TUNnelVRing = 100
```

When using source routing, the internetwork becomes a virtual ring. If your end systems are using token ring source routing, the bridge/router and the IP tunnel appear to the end systems as a source route bridge with a token ring network attached to the other side of the bridge/router. The default ring number of this virtual ring is decimal 92.



This virtual ring number must match on all peer bridge/routers used for data link switching and must be unique within the token ring network. It also will minimize the risk of topology loops.

- 5 Configure an IP address to connect traffic to and from the router.

The address must be one that has been defined in the router, and will be the only address used for data link switching. To map the specified DLSw tunnel to the local IP address of the central site bridge/router, enter:

```
SETDefault -DLSw Interface = 129.213.1.1
```



All Internet addresses for connected bridge/routers must be known in the local bridge/router's routing table, either dynamically through RIP or OSPF, or statically configured in the IP routing tables.

- 6 Set the central site bridge/router to accept all DLSw connection requests (including requests from bridge/routers that are not configured as data link tunnel peers) by entering:

```
SETDefault -DLSw MOde = NonSecure
```

The difference between this host configuration and the example configuration for SNA shown in Figure 245 is that the host-located data link switch does not need to be configured with the IP address of its partners.

Setting Up DLSw Security Access Filters

You can configure data link switching with additional security beyond what is defined with DLSw peers and known IP addresses. With the `-DLSw AccessAct` parameter, you can configure the media address you are permitting access to for SNA traffic, or for NetBIOS traffic you can configure specific NetBIOS names of devices you are permitting access to.

Setting Up Filters for SNA Traffic

The following examples of setting up security access for SNA traffic see Figure 245. Examples 1 and 2 configure bridge/router B for security access.

Example 1 If you want to prevent PC1 from accessing the host, at the bridge/router on network B, enter:

```
SETDefault -DLSw AccessAct = RemoteSnaDiscard  
ADD !1 -DLSw SnaRemAccess 00608C26C1B5 ffffffff 10005A265BED  
ffffffff
```

If you want to prevent PC1 from accessing any remote system at the bridge/router on network B, enter:

```
SETDefault -DLSw AccessAct = RemoteSnaDiscard  
ADD !1 -DLSw SnaRemAccess 00608C26C1B5 ffffffff 000000000000  
000000000000
```

Example 2 If you want to allow only PC1 access to the host, but want to restrict access to all other systems at the bridge/router on network B, enter:

```
SETDefault -DLSw AccessAct = RemoteSnaForward  
ADD !1 -DLSw SnaRemAccess 00608C26C1B5 ffffffff 10005A265BED  
ffffffff
```

Examples 3 and 4 configure bridge/router A for security access.

Example 3 If you want to prevent PC1 from accessing the host, at bridge/router A, enter:

```
SETDefault -DLSw AccessAct = LocalSnaDiscard
```

```
ADD !1 -DLsw SnaLocalAccess 00608C26C1B5 ffffffff 10005A265BED
fffffff
```

If you want to prevent PC1 from accessing any system attached to bridge/router B, at bridge/router A, enter:

```
SETDefault -DLsw AccessAct = LocalSnaDiscard
ADD !1 -DLsw SnaLocalAccess 00608C26C1B5 ffffffff 000000000000
000000000000
```

Example 4 If you want to allow only PC1 to access the host, but restrict access for all other local systems, at bridge/router A, enter:

```
SETDefault -DLsw AccessAct = LocalSnaForward
ADD !1 -DLsw SnaLocalAccess 00608C26C1B5 ffffffff 10005A265BED
fffffff
```

Setting Up Filters for NetBIOS Traffic

The following examples of setting up security access for NetBIOS traffic see Figure 246. Examples 1 and 2 configure bridge/router B for security access.

Example 1 If you want to prevent PC001 from accessing the LAN server LS0001, at the bridge/router on network B, enter:

```
SETDefault -DLsw AccessAct = RemoteNBDiscard
ADD !1 -DLsw NBRemAccess PC0001 LS001
```

Example 2 If you want to allow only PC0001 access to LAN server LS0001, but want to restrict access to all other systems, at the bridge/router on network B, enter:

```
SETDefault -DLsw AccessAct = RemoteNBForward
ADD !1 -DLsw NBRemAccess PC0001 LS0001
```

Examples 3 and 4 configure bridge/router A for security access.

Example 3 If you want to prevent PC0001 from accessing LS0001, at bridge/router A, enter:

```
SETDefault -DLsw AccessAct = LocalNBDiscard
ADD !1 -DLsw NBLocalAccess PC0001 LS0001
```

Example 4 If you want to allow only PC0001 to access LS0001, but restrict access for all other local systems, at bridge/router A, enter:

```
SETDefault -DLsw AccessAct = LocalNBForward
ADD !1 -DLsw NBLocalAccess PC0001 LS0001
```

Disabling Data Link Switched Connections

You can disable tunneled data link switch peer connections for a specific peer by tunneling to and from an internetwork, or disabling all tunneling on the local bridge/router.

To disable tunneling from a switch to a peer network, enter:

```
SETDefault !1 -DLsw PEer = 129.213.1.2 Disable
```

This command disables a connection to a peer data link switch.

To disable all tunneling on the bridge/router, enter:

```
SETDefault -DLsw Interface = 0.0.0.0
```


Configuring Statically Defined Media Addresses

If your installation requires multiple DLSw tunnels, you can configure your data link switch connections to use statically defined media addresses. For example, to configure bridge/router A in the SNA example shown in Figure 245 with the host media address, enter the following command using noncanonical format for the address:

```
ADD !1 -DLSw PeerMacAdd 10005A265BED
```

Explorer type frames are then sent to the one predefined DLSw peer address.

Configuring Statically Defined NetBIOS Names

If your installation has routers with multiple DLSw tunnels, you can configure your data link switch connections to use statically defined NetBIOS names. For example, to configure bridge/router A in the NetBIOS example shown in Figure 246 with the host name, enter:

```
ADD !1 -DLSw PeerNBName LANSERVER1
```

This setting ensures that Name Query frames are not broadcast to all DLSw peer addresses, but are sent only to the predefined DLSw peer.

The section describes how to increase performance and reduce NetBIOS broadcasts.

The 3Com DLSw router at the receiving end of a NetBIOS broadcast sends only one NetBIOS broadcast across the data link switch. The remote DLSw Peer router receives the NetBIOS broadcast and resends this same frame as many times as are defined by the NBBdcastResend parameter at the configured time interval.

If you want to change the values for NetBIOS broadcasts, enter:

```
SETDefault -DLSw NBBcastResend = 5
SETDefault -DLSw NBBcastTimeout = 2
```

NBBcastResend and NBBcastTimeout are independent of each other. Setting one parameter does not effect the other parameter.

Prioritizing DLSw Traffic

This section describes how to assign priorities and allocate bandwidth percentage to traffic from individual workstations, allowing you to give higher priority to mission-critical applications. The address of workstations and host computers in this section are used for example purposes only. Be sure to use the correct addresses for your network.



If the physical port that DLSw is using is being used by another protocol, you also may want to configure data prioritization. For information about how to configure data prioritization, see the Prioritizing Multiprotocol Data chapter.

How Prioritization and Bandwidth Allocation Work

DLSw allows you to allocate bandwidth to traffic coming from individual workstations on network segments directly attached to a NETBuilder II bridge/router. In addition, DLSw allows you to assign priority to traffic coming from workstations. By assigning priority, you specify the order in which packets from workstations are placed on the link between NETBuilder and WAN services. This effects traffic delays but not traffic throughput. By allocating bandwidth, you specify how much link bandwidth the packets receive.

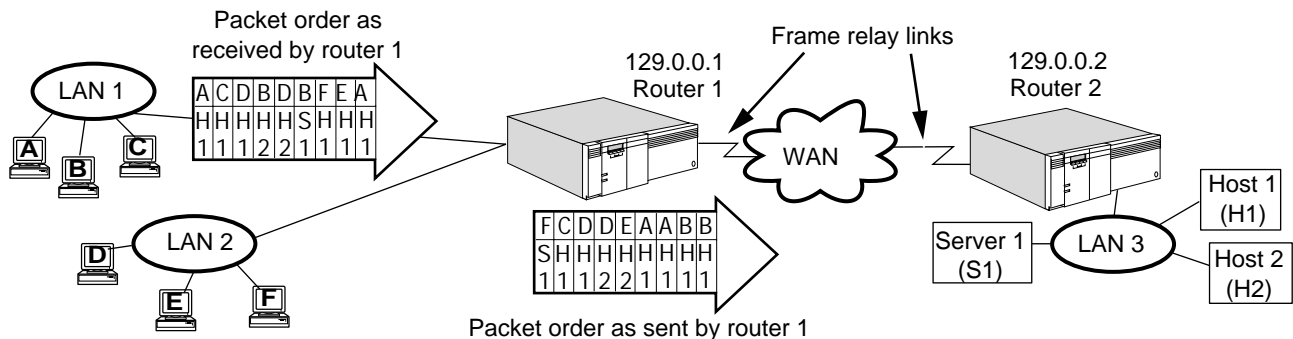
Example 1 This example illustrates how prioritization and bandwidth allocation work together.

Workstation X is set to High priority and 20% bandwidth. Workstation Y is set to Low priority and 80% bandwidth. Both workstations are sending many packets to the same tunnel. Of every ten packets the tunnel sends, the first two are from workstation X and the last eight are from workstation Y.

Example 2 Figure 248 shows two NETBuilder II bridge/routers, router 1 and router 2, as DLSw peers connected by a Frame Relay circuit.

To use the prioritization feature with this network, enter the local workstation's MAC address, service access point (SAP), or LU address identifier. Enter the same information for the workstation's remote session partner. The terms local and remote refer to the router from which you are configuring. For example, in Figure 248, you are configuring from router 1, and the addresses for its devices are local. The addresses for devices attached to router 2 are remote. In the figure, each letter represents a different MAC address.

Figure 248 DLSw Prioritization and Bandwidth Allocation Example



There are six workstations, A through F, connected to router 1 through LAN 1 and LAN 2. There also are two SNA hosts, host 1 and host 2, and one NetBIOS server, server 1, connected to router 2. The following is the prioritization criteria defined for DLSw traffic going from router 1 to router 2:

- SNA traffic from workstation A to host 1 has a medium priority and 20% of the link bandwidth between router 1 and the Frame Relay service provider.
- SNA traffic from workstation B to host 1 has a high priority and 30% of the link bandwidth between router 1 and the Frame Relay service provider.
- SNA traffic from workstations C, D, E, and F to host 1, host 2, and server 1 has a medium priority and 40% of the link bandwidth between router 1 and the Frame Relay service provider.
- NetBIOS traffic from workstation F to server 1 has a low priority and 10% of the link bandwidth between router 1 and the Frame Relay service provider.
- As Figure 248 shows, packets coming from all the workstations get reordered for output on the Frame Relay link based on assigned priorities. For example, router 1 receives some packets from workstation F before it receives some packets from workstation A. However, because A has a high priority and F a low priority, A's packets are sent first because F's priority is lower than the other workstations. F's packets are sent last.



DLSw does not waste tunnel bandwidth. Bandwidth not allocated can be used by any workstations routed through the same tunnel.

Configuring Bandwidth Allocations and Priorities

DLSw allows you to allocate connection bandwidth and assign priorities to traffic from individual workstations. This section describes how to configure the example in Figure 248. The addresses of workstations and host computers in this section are not the addresses you are going to use for your network. Be sure to use the correct addresses for your network.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the token ring LAN as described in the Configuring Source Route Bridging chapter.
- Configure the IP addressing and IP routing protocols on the appropriate ports in the Configuring IP Routing chapter.
- Obtain the IP addresses for both bridge/routers on either sides of the TCP/IP connection.
- Configure DLSw for both bridge/routers.
- Set the default DLSw mode to DefaultPRioritized if most of your tunnels are going to be prioritized.
- Add a PEer definition for the remote router and set it to Disable. In Figure 248, tunnel ID number 1 is used. When you configure your network, you can use any number between 1 and 256 for tunnel numbers. The peer needs to be defined PRioritized if the default mode is not set to DefaultPRioritized.

Procedure

To configure the example in Figure 248, follow these steps:

- 1 Add a prioritization criterion for workstation A and its session partner, host 1, to the DLSw prioritization database, by entering:

```
ADD !3 -DLSw PRiorityCRiteria 1 20 Medium A SNA H1 SNA
```

Workstation A and host 1 are added to instance ID 3 in the DLSw prioritization database. In addition, workstation A's packets are allocated 20% of the tunnel bandwidth of tunnel ID 1 on router 1 and given a medium priority. The SAP address for workstation A and host 1 is SNA. In this command, the letter's A and H1 represent real MAC addresses; this also applies to the letters in the commands entered in steps 2 and 3.

- 2 Add a prioritization criterion for workstation B and its session partner, host 2, to the DLSw prioritization database, by entering:

```
ADD !4 -DLSw PRiorityCRiteria 1 30 High B SNA H2 SNA
```

- 3 Add a prioritization criterion for workstation F and its session partner, server 1, to the DLSw prioritization database, by entering:

```
ADD !5 -DLSw PRiorityCRiteria 1 10 Low F NB S1 NB
```

- 4 Add a prioritization criteria for all remaining session partners to the DLSw prioritization database, by entering:

```
ADD !6 -DLSw PriorityCriteria 1 40 Medium * SNA * SNA
```

- 5 Enable the connection between the devices attached to router 1 and the data link switch by entering:

```
SETDefault -DLSw PEer = 129.0.0.2 Enable
```

After you configure session pairs from router 1, you need to configure session pairs from router 2 if you have set the -DLSw MObde parameter to SECure.

Examples of Other Commands

- Example 1* To delete an instance ID from the DLSw prioritization database, enter:

```
DELEte !3 -DLSw PriorityCriteria 1
```

Instance ID 3 is deleted from tunnel ID 1.



CAUTION: *This command deletes every attribute defined for each device associated with the instance ID and tunnel ID.*

- Example 2* To display information in the DLSw prioritization database, enter:

```
SHow -DLSw PriorityCriteria 1
```

All the instance IDs associated with tunnel ID 1 in this example are displayed.

- Example 3* To change prioritization criterion number 3 to 60% bandwidth, enter:

```
SETDefault !3 -DLSw PriorityCriteria 1 60
```

If the percentages do not add up to 100%, DLSw normalizes them to 100%.

All devices connected to router 1 that also are associated with instance ID 3 are now allocated 43% of the tunnel bandwidth. DLSw performs the following normalization calculation: $60\% / (60\% + 30\% + 10\% + 40\%) = 43\%$.

- Example 4* To display prioritized statistics for tunnels on the local router, enter:

```
SHow -DLSw PrioritySTATistics
```

The following display is an example of these statistics:

```
-----DLSw PrioritizationSTATistics 192.0.60.10-----
Tid CurBw   BytesPassed
1   8000     16073
-----CriteriaStatistics-----
Cid Config% History% BytesPassed HoldQSize
1   30      0         0         0
0
2   20      34        5484      0
0
3   30      56        9035      0
0
4   20      10        1730      0
0
33  0        0         0         0
0
```

- Example 5* To reset statistics for tunnels on the remote router, enter:

```
FLush -DLSw PrioritySTATistics
```

Statistics for router 2 tunnels are cleared.

For more information about prioritizing tunnel traffic., see the DLSw Service Parameters chapter in *Reference for Enterprise OS Software*.

Prioritizing DLSw Packets

To set the traffic priority of DLSw packets, use:

```
SETDefault -LLC2 TUNnelPRiority = <H | M | L | DEFault>
```

Using this parameter, you set the priority of the packets to high, medium, or low. If this parameter is set to default, the system uses the -IP QueuePriority setting. For more information about the TUNnelPRiority parameter, see the LLC2 Service Parameters chapter in *Reference for Enterprise OS Software*.

The priority you set using the -LLC2 TUNnelPRiority parameter is different from the priority criteria set using the -DLSw PriorityCriteria parameter. The latter parameter only sets the criteria for prioritizing SNA traffic versus NetBIOS traffic.



When setting prioritization for DLSw packets, UDP explorer frames are automatically set to high priority regardless of the -LLC2 TUNnelPRiority parameter setting. The priority of all other types of UDP packets is set using -LLC2 TUNnelPRiority.

Circuit Balancing

This section describes how to configure DLSw to distribute sessions evenly over multiple DLSw connections and use alternate routes (tunnel paths) for sessions.



If DLSw multicast is being used, circuit balancing is not necessary.

How Circuit Balancing Works

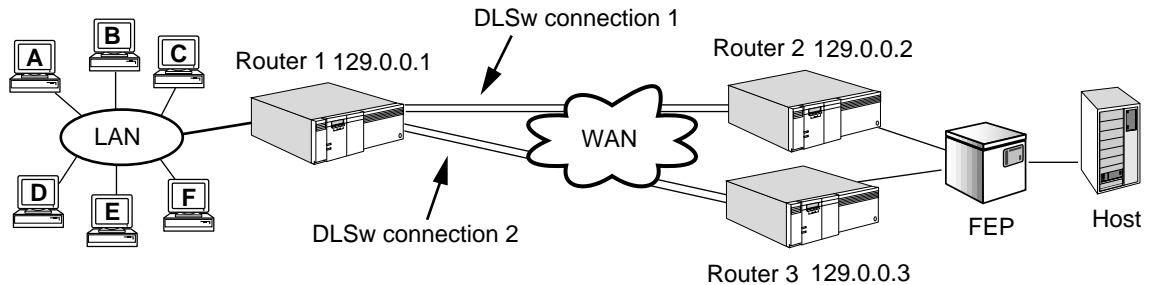
The circuit balancing feature of DLSw allows you to use more than one route between end-stations. When you enable circuit balancing, DLSw considers all available routes between end-stations before assigning a session to a tunnel. DLSw also distributes sessions evenly across all available routes. For example, if there are two routes, and one route has two sessions and the other has three, DLSw assigns the next incoming session to the first route. If a connection fails, DLSw disruptively reroutes end-station and host sessions to an available route (users have to reestablish their sessions with host applications).

Figure 249 shows a SuperStack II NETBuilder bridge/router (router 1) with one token ring LAN attached. The LAN also has six workstations attached. router 1 has WAN connections to two NETBuilder II bridge/routers (router 2 and router 3) attached to a front-end processor (FEP) at a host site. Traffic between end-stations (the workstations) and the host travels through DLSw tunnels, and the circuit balancing feature of DLSw is enabled.

When router 1 is configured for circuit balancing, DLSw distributes sessions evenly between Connection 1 and Connection 2. If one of the connections fails, DLSw

disruptively reroutes sessions between workstations on the LAN and the host by moving them to the other tunnel.

Figure 249 Circuit Balancing Example



For circuit balancing to function properly, WAN links must be the same speed. If the WAN links shown in the figure are different speeds (for example, one link is T1 and the other is 64K), then the router with circuit balancing learns the route from the T1 link before the learning the route from the 64K link. All circuits are directed to the DLSw connection on the T1 link instead of being distributed on both the 64K and T1 DLSw connections. Only after alternate routes are in the cache of the circuit balanced router, is the subsequent session establishment balanced (for example, an SNA session to the same MAC address destination is deactivated and then reactivated again).

Configuring Circuit Balancing

This section describes how to configure the example in Figure 249.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to each bridge/router with Network Manager privilege.
- Set up the ports and paths of the bridge/routers according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the token ring LAN as described in the Configuring Source Route Bridging chapter.
- Configure the IP addressing and IP routing protocols on the appropriate ports as described in the Configuring IP Routing chapter.
- Obtain the IP addresses for the three bridge/routers.
- Configure DLSw for the three bridge/routers.
- Set the default DLSw mode to DefaultPRioritized if most of your tunnels are going to be prioritized.
- Add a -DLSw PEer definition for the remote router and one for each host router, and set them to Disable. The peer needs to be defined NoPRioritized if the default mode is not set to DefaultPRioritized.

Procedure

To configure circuit balancing for SNA bridge/router 1, see Figure 249 and follow these steps from the router 1 console. Be sure to use the addresses and commands appropriate for your network.

- 1 Enable circuit balancing for traffic between router 1 and router 2, and between router 1 and router 3 by entering:

```
SETDefault -DLsw CircuitBal = Enable
```



Unless you specify a <cache refresh timeout> value in the SETDefault command, DLSw defaults to 60 minutes. Cache refresh timeout is the interval between each route discovery broadcast.

- 2 Confirm that circuit balancing is enabled by entering:

```
SHow -DLsw CircuitBal
```

Examples of Other Circuit Balancing Commands

Example 1 To set the interval between each route discovery broadcasting between router 1 and router 2 to 100 minutes, enter:

```
SetDefault -DLsw CircuitBal = Enable 100
```

Example 2 To prevent DLSw from assigning any new circuits (sessions) to Tunnel 1, enter:

```
SETDefault !1 -DLsw PEer = 129.0.0.2 Enable
SET -DLsw CONNections = 129.0.0.2 Quiesce
```

Example 3 To prevent DLSw from sending broadcast or explorer packets on Tunnel 2, enter:

```
SETDefault !2 -DLsw PEer = 129.0.0.3 Enable NoBroadcast
```

Router 1 still accepts and answers explorer packets from router 2 and router 3. The NoBroadcast setting prevents circuits from being initiated from this side.

Configuring Local Switching and Port Groups

You can use local switching and port groups to design DLSw topologies over remote connections for the following situations:

- When you need to translate from one type data link control (such as LLC2) to a different type (such as SDLC), and when you need to concentrate traffic from multiple input ports to one output port locally. This translation is done implicitly and no user configuration is required. See "Using Local Switching to Translate Different DLC Traffic Types" next.
- When you need to convert LAN LLC2 traffic at a branch office to Frame Relay LLC2 traffic (conforming to RFC 1490) that feeds into a remote NETBuilder bridge/router at a regional office, which in turn sends the traffic over DLSw connections to another bridge/router at the central site. This method reduces the number of incoming DLSw connections to the central site. See "Configuring Port Groups for Funneling Many Remote Connections Into Fewer DLSw Connections" later in this chapter.



Local switch port grouping is supported over Frame Relay links only. Also, local switch port grouping cannot be used for BSC traffic.

Using Local Switching to Translate Different DLC Traffic Types

You can configure local switching port groups to funnel connections from many LANs into a single bridge/router and in turn funnel these multiple connections

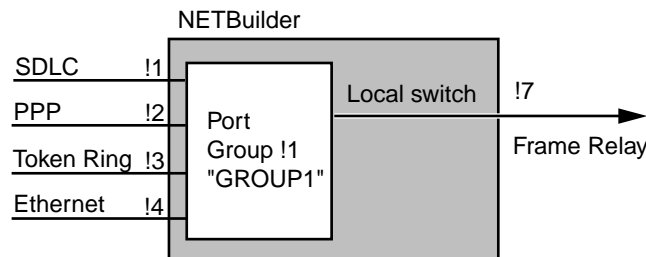
through a single data link switch to reach the central site host. With this capability, you can switch incoming traffic of one type to outgoing traffic of the same type or another type on the same bridge/router.

Local switching port groups can be used in specific network topologies where Frame Relay Access Device (FRAD) functionality is desired but is not efficient. For example, port groups can be used in configurations in which IBM traffic is forwarded from an LLC2 or RFC 1490 domain to a Frame Relay circuit that connects to a central site in RFC 1490 format but without the MAC address translation. The central site in this configuration usually hosts so many stations that configuring each remote MAC address into the mapping table is impractical. If you use the FRAD capability, you are required to configure these remote MAC addresses. Local switching port groups enable you to set up such a large network without having to configure hundreds of remote MAC addresses. For more information about FRAD and BAN, see the Configuring Frame Relay Access Device Support for SNA chapter.

Port groups configured using this feature are known as explicit port groups. Ports defined as SDLC, FRAD, BAN, or LLC2 (for ports that are LAN encapsulated) are known as implicit port groups. The local switching feature enables you to switch traffic from a port group to other port groups.

Figure 250 is an example of configuring port grouping to enable local switching on the bridge/router. In the figure, ports 1 through 4 are incoming ports over a variety of media. These four ports are grouped into port group 1 on the bridge/router; all incoming traffic over the four ports are switched to Frame Relay and are then sent to the Frame Relay WAN over port 7.

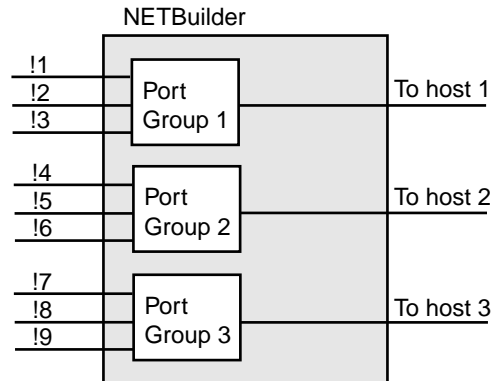
Figure 250 Port Group Local Switching



```
ADD !1 -DLSw PortGroup 1 2 3 4 "GROUP1"
```


You can configure up to eight external port groups on a single bridge/router. Figure 251 is an example of multiple port groups on a bridge/router, with each port group forwarding the traffic from its port group to a different host.

Figure 251 Multiple Port Groups on a Bridge/Router

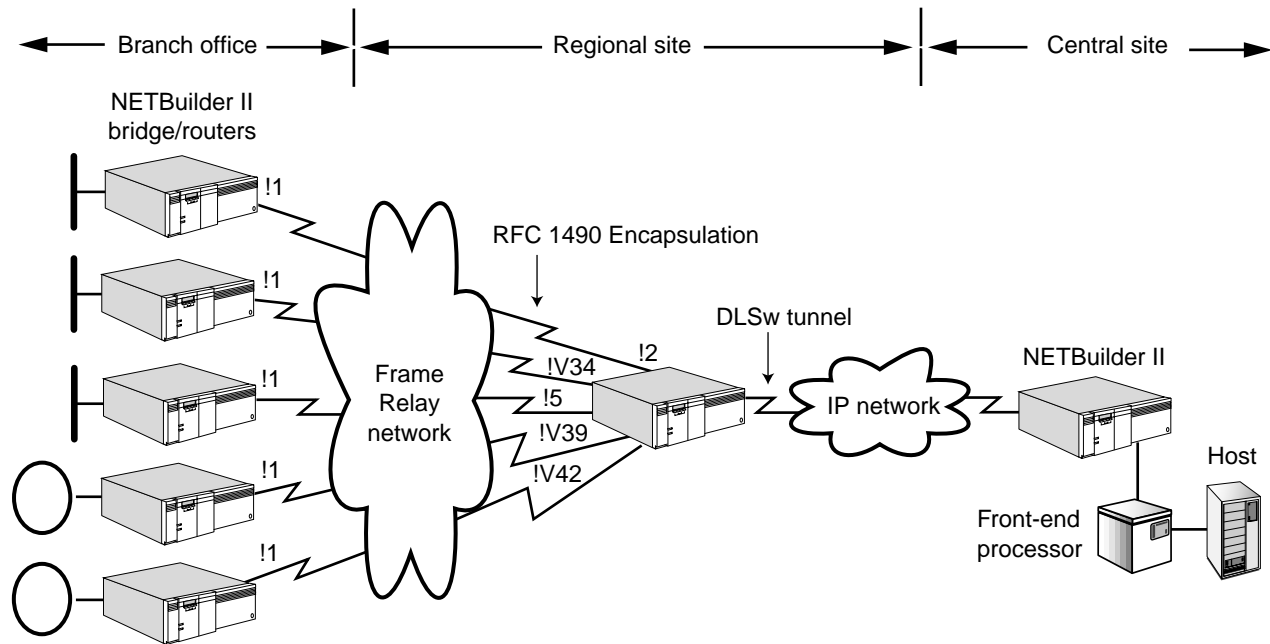


Configuring Port Groups for Funneling Many Remote Connections Into Fewer DLSw Connections

Using port groups, you can reduce the number of Frame Relay connections needed, which enables greater scalability for larger networks. Figure 252 is an example of how port grouping can be used to funnel many connections on a large network down to one. In the example, many branch office LANs are connected across a Frame Relay network to a NETBuilder bridge/router at a regional site; the regional site in turn is connected to the host front-end-processor at the central site across another Frame Relay network. By setting up the port groups, you can group the multiple LAN connections from the branch offices and funnel them to the central site across the single data link switch tunnel.

Using this approach, you can avoid having hundreds of DLSw tunnels from each branch office terminating at the central site bridge/router. By combining all SNA traffic from the branch offices into a single Frame Relay DLSw tunnel, you can greatly reduce the number of tunnels at the central site, enabling greater scalability for large networks.

Figure 252 DLSw Port Group Topology



You configure the port groups on the NETBuilder bridge/router at each branch office. No special configuration is required at the regional site other than the normal DLSw and port and path configurations.



You cannot have local switch port grouping enabled while bridging is enabled. Before configuring port groups, make sure bridging is disabled.

To configure a port group on the branch office NETBuilder bridge/router in the example, follow these steps on each branch office bridge/router:

- 3 Enable LLC2 on port 1 by entering:

```
SETDefault !1 -LLC2 CONTROL = Enable
```

- 4 On the branch office NETBuilder bridge/router, enable source route bridging on port 1 by entering:

```
SETDefault !1 -SR SrcRouBridge = SrcRouBridge
```

- 5 Define the regional office NETBuilder bridge/router as the DLCI neighbor using:

```
ADD !<port> -BRidge DlcNeighbor = <dlci> (16-991)
```

For example, to define the DLCI neighbor as 20 for port 1, enter:

```
ADD !1 -BRidge DlcNeighbor = 20
```

When you configure the regional office NETBuilder bridge/router, you must also define the DLCI neighbor as 20, so the two bridge/routers can send and receive traffic over Frame Relay.

6 Set the DLCI throughput using:

```
SETDefault !<port> -FR DLCIR = <dlci> <cir>
```

Using this command, define the throughput using the <cir> value based on your service provider's requirements. For example, to define this parameter for port 1 for DLCI number 20 with a <cir> value of 64 (for 64 kbps), enter:

```
SETDefault !1 -FR DLCIR = 20 64
```

When you configure the regional office NETBuilder bridge/router, you must also define this parameter with the same value so the two bridge/routers can send and receive traffic over Frame Relay.

7 Define the port group using:

```
ADD !<port_group_id> -DLSw PortGroup <port> [,...] ["<string>"]
```

For example, to create port group 1 and assign port 1 to it, enter:

```
ADD !1 -DLSw PortGroup 1
```

Using the PortGroup parameter, you can assign up to 16 ports to a port group, and you can also assign a string to the port group. For example, to assign ports 2, 3, 4, and 5 to the port group and assign the string PG1 to it, enter:

```
ADD !1 -DLSw PortGroup 2, 3, 4, 5 "PG1"
```

8 Repeat the previous steps for each branch office bridge/router that will be accessing the same host, assigning the specific ports as necessary.

The port group number only needs to be unique on the local bridge/router. The port group number does not need to match on other bridge/routers.

Table 63 lists the commands you need to enter on both the branch office NETBuilder bridge/router and the bridge/router at the regional site for port groups to work.

Table 63 Commands to Configure Local Switch Port Groups on Both Bridge/Routers

| Commands Entered on the Branch Office Bridge/Routers | Commands Entered on the Regional Office Bridge/Router (entered on the WAN ports to the branch office bridge/routers) |
|--|--|
| SETDefault !1 -LLC2 CONTROL = Enable | SETDefault !<port> -LLC2 CONTROL = Enable |
| SETDefault !1 -SR SrcRouBridge = SrcRouBridge | SETDefault !<port> -SR SrcRouBridge = SrcRouBridge |
| ADD !1 -BRidge DlcINeighbor = 20 | ADD !<port> -BRidge DlcINeighbor = 20 |
| SETDefault !1 -FR DLCIR = 20 64 | SETDefault !<port> -FR DLCIR = 20 64 |
| ADD !1 -DLSw PortGroup 2, 3, 4, 5 "PG1" | |

The following restrictions relate to the use of port groups:

- You cannot use redundant links with port groups.
- SHDLC links are not supported on port groups.

To delete ports in a port group or an entire port group, use:

```
DElete !<port_group_id> -DLSw PortGroup [<port> [,...] | ALL]
```

For example, to delete ports 4 and 5 in port group 1, enter:

```
DElete !1 -DLSw PortGroup 4, 5
```

To delete all ports in port group 1 (and thus delete port group 1), enter:

```
DElete !1 -DLSw PortGroup ALL
```

Network Design Issues for Port Grouping

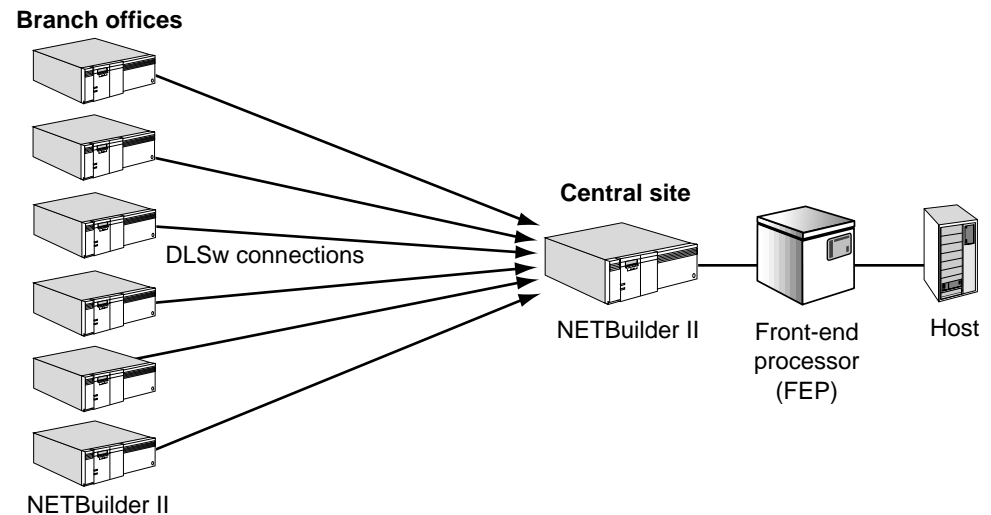
You can use port grouping to solve the following DLSw network design issues:

- Scaling large DLSw networks
- Scaling large meshed DLSw networks

The following sections describe these issues.

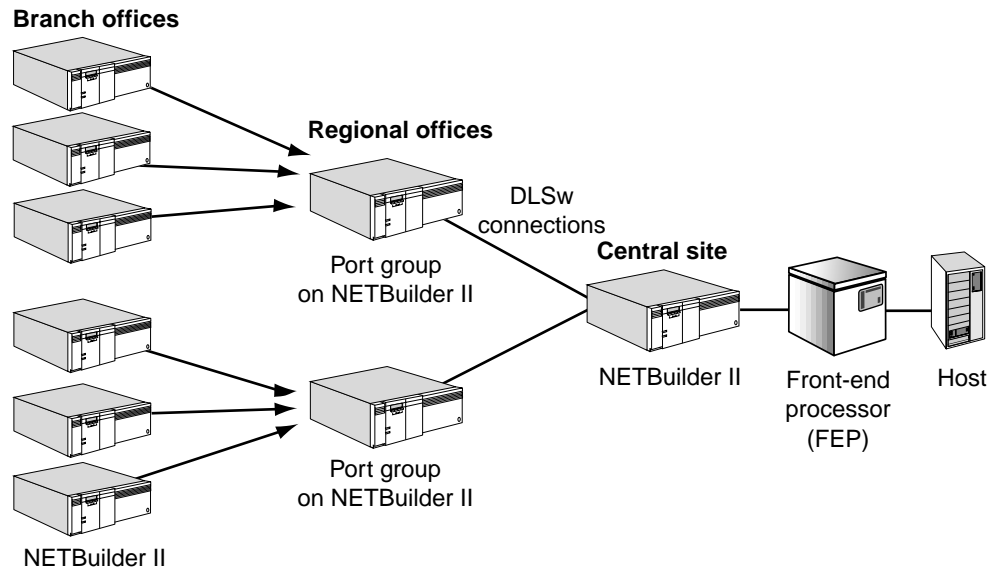
Using Port Groups to Scale Large DLSw Networks Figure 253 is an example in which NETBuilder bridge/routers at six separate branch offices each have a DLSw connection across an IP network into a NETBuilder II bridge/router at a central site. Because there are six DLSw connections, the central site must deal with the overhead and processing for each connection.

Figure 253 DLSw Connections to Remote Offices (Before Port Grouping)



In Figure 254, port groups have been configured on NETBuilder bridge/routers at regional offices. Each port group has three remote site branch offices assigned to it, with the three remote connections funneled through a single DLSw connection to the central site. By assigning port groups in this way, you can reduce the number of incoming DLSw connections to the central site from six to two.

Figure 254 DLSw Connections to Remote Offices (After Port Grouping)



Using Port Groups to Scale DLSw Meshed Networks Figure 255 is an example of a DLSw meshed network in which there are bridge/routers at nine remote sites, each configured with DLSw connections so that every site can communicate directly with every other site. Such meshed topologies create additional overhead of large numbers of Frame Relay circuits and TCP connections and create problems with topology update broadcasts.

Figure 255 DLSw Meshed Network (Before Port Grouping)

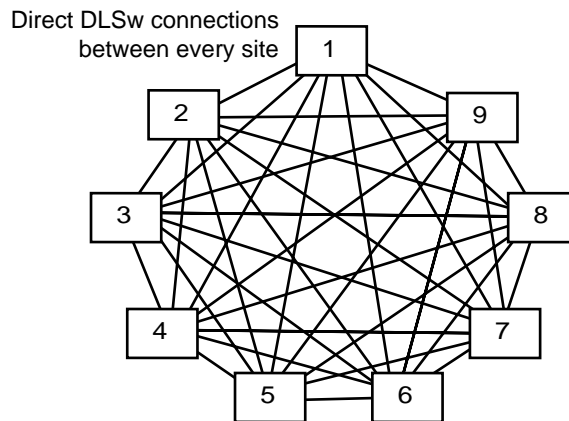
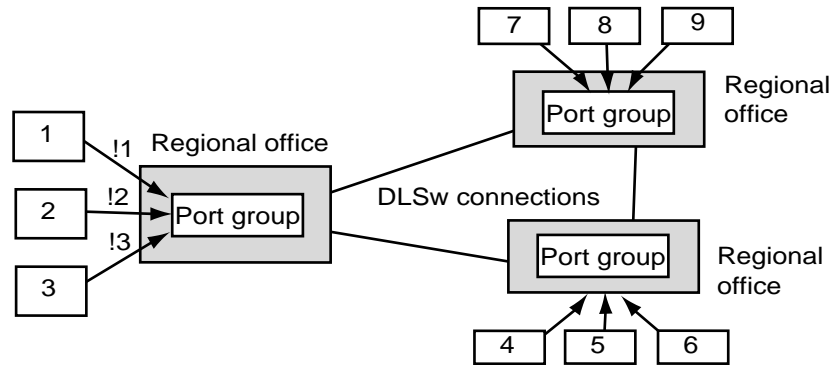


Figure 256 shows the same meshed network with port groups configured at intermediate regional offices. By configuring port groups on each of the remote sites funneling into three regional offices, each remote site can connect with every other remote site. By assigning port groups in this way, you can reduce the number of DLSw connections from 35 to three. On the regional office bridge/routers, you must either have bridging enabled, or you can configure a port

group on each regional office bridge/router for the ports incoming from the remote sites.

Figure 256 DLSw Meshed Network (After Port Grouping)

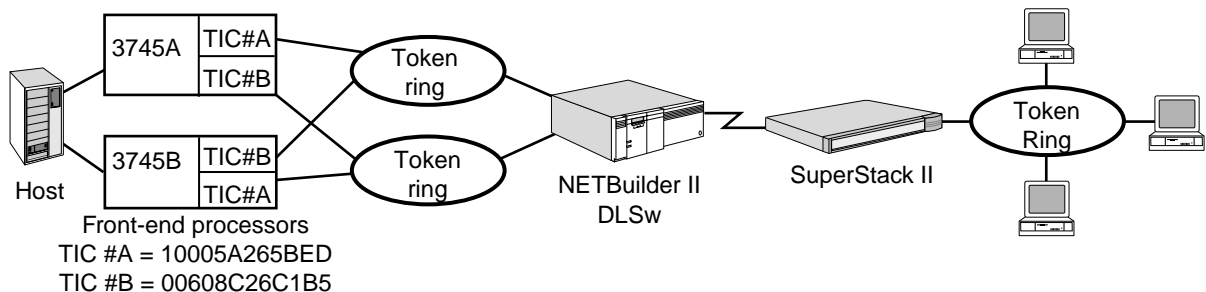


Configuring DLSw for Dual-TIC Topologies

Host topologies are often designed so that the same MAC address is assigned to multiple token ring interface Cards (TICs) on front-end-processors. This configuration, referred to as a dual-TIC topology, provides greater backup, redundancy, and load balancing across the dual interface cards. The NETBuilder DLSw implementation supports dual-TIC topologies in source-routed environments.

Figure 257 is an example in which dual TICs are set up on two token rings. The TICs on the front-end-processors have redundant MAC addresses. For example, the MAC address 10005A265BED is mapped to TIC #A on both 3745A and to TIC #A on 3745B, while MAC address 00608C26C1B5 is mapped to TIC #B on both front end processors.

Figure 257 DLSw in a Source Route Dual-TIC Topology



In this configuration, the NETBuilder II bridge/router supports the dual-TIC environment. No special configuration is required to support dual-TIC except for the following steps:

- You must configure ring numbers for the token rings accessing the front-end-processors
- You must turn off transparent bridging (use source route bridging only)

For more information, see the Configuring Source Route Bridging chapter.

Converting SNA Alerts to SNMP Traps

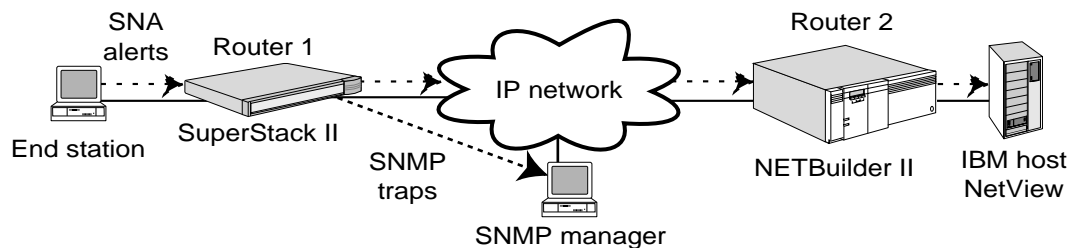
This section describes how the SnaAlertsToTraps feature of DLSw converts SNA alerts to SNMP traps so that SNMP managers, such as SunNet Manager, NetView AIX, or HP OpenView, can process them. When SNA devices detect a problem, they can send SNA alerts to a focal point (usually NetView) where they are processed and displayed to an operator. The alerts contain information describing the problem and possible actions to be taken.

How SNA-Alerts-To-Traps Works

DLSw allows you to interconnect devices such as OS/2 workstations and 3174 cluster controllers to SNA hosts using NETBuilder II bridge/routers. The SnaAlertsToTraps feature of DLSw enables SNMP management platforms to manage SNA devices (end-stations) by converting their SNA alerts to SNMP traps and sending the traps to the SNMP manager.

Figure 258 shows an end-station and an IBM host connected by a SuperStack II bridge/router (router 1) and a NETBuilder II bridge/router (router 2) over an IP network. The end-station sends SNA alerts to router 1, which passes them to the IBM host, where NetView processes and displays them to an operator. router 1 converts the SNA alerts to SNMP traps and sends the traps to the SNMP manager.

Figure 258 SnaAlertsToTraps Example



Configuring SnaAlertsToTraps

To configure the SnaAlertsToTraps feature, follow these steps from the SuperStack II console:

- 1 Set the trap option for the SNMP Service by entering:

```
SETDefault -SNMP CONTROL = Trap
```



The SnaAlertsToTraps feature does not work unless trap is set using the SNMP Service. For more information about how to configure the NETBuilder II bridge/router so that it can be controlled by an SNMP manager, see the Network Management chapter and the SNMP Service Parameters chapter in Reference for Enterprise OS Software.

- 2 Enable the SnaAlertsToTraps feature by entering:

```
SETDefault -DLSw SnaAlertsToTraps = Send
```

When you enable SnaAlertsToTraps, the SuperStack II bridge/router processes SNA alerts for every attached LU and PU.

You can use two other values instead of Send. The SendAlert value encapsulates the entire SNA alert (the Network Management Vector Transport (NMVT)) inside an SNMP trap protocol data unit (PDU), and sends it to the SNMP manager. The Disabled value tells the SuperStack II bridge/router to ignore all SNA alerts.

To verify the current state of the SnaAlertsToTraps feature, enter:

```
Show -DLSw SnaAlertsToTraps
```

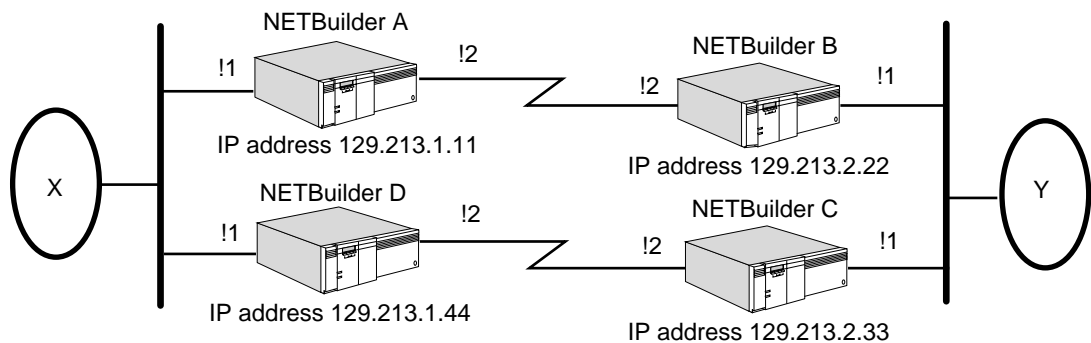
A display indicates whether the SnaAlertsToTraps feature is enabled.

For more information about configuring the SnaAlertsToTraps feature, see the DLSw Service Parameters chapter in *Reference for Enterprise OS Software*.

Enabling DLSw Loop Detection

Because DLSw is a tunnelling protocol that forwards session initiation frames out to every DLSw tunnel in much the same way bridges perform forwarding, DLSw is prone to the same looping problem experienced by bridges. In a source routing environment, looping is prevented by checking the route-information-field (assuming that all ring numbers are uniquely configured). But in a transparent bridging environment as shown in Figure 259, test frames sent out by X are forwarded to Y through both the connection between NETBuilder A and NETBuilder B (note that this connection is also referred to as a tunnel) and the connection between NETBuilder C and NETBuilder D and the test frames are also looped back to the LAN where X resides. When Y responds to the test frame, the response gets forwarded back to X through both connections again. Because of this loop, two DLSw circuits are established here, which causes every data packet sent by X to Y to be received twice by Y. Eventually the session is cancelled due to LLC2 protocol errors.

Figure 259 Simple Transparent Bridging Configuration



When the `DlswLoopDetect` command is entered, for example in bridge/router labeled NETBuilder A, a special testloop frame with a reserved, well-known destination MAC address (DA) with a reserved and a well-known source MAC address (SA) is created and propagated through the DLSw logic as if it is received on one of the ports, in this case port !1. When the testloop frame is processed, a CANUREACH SSP packet is sent to both the connection between NETBuilder A and NETBuilder B and the connection between NETBuilder D and NETBuilder C, in this example, the connection between NETBuilder A and NETBuilder B (tunnel NETBuilder A-NETBuilder B) and the connection between NETBuilder D and NETBuilder C (tunnel NETBuilder D-NETBuilder C).

The CANUREACH packet behaves the same for each of the tunnels described. In addition to having a special unicast address as DA, each testloop frame also carries the route information indicating that it originates from NETBuilder A.

When the remote DLSw node (NETBuilder B) receives the CANUREACH frame, it recognizes it as a testloop frame based on the special DA carried in the SSP header. The remote DLSw node (NETBuilder B in this example) adds new routing information to the frame and forwards it to its ports as TEST frames. If loops do exist, this testloop frame is eventually looped back to NETBuilder A and either is received on a legacy port or on a tunnel port. Again it is recognized as a testloop frame because of the special DA. In addition, NETBuilder A knows that this frame

was originated by itself by checking the route information. When a testloop frame is received back by the originating bridge/router, the loop and its route information is displayed to the user.

Initiating Loop Detection

To invoke DLSw loop detection on NETBuilder A, at the NETBuilder prompt, enter:

```
DlswLoopDetect
```

By default, the command applies to all tunnels, the source address is the reserved unicast address, and the operation timeouts after 20 seconds. You can optionally specify which tunnels to observe, a different source address, and how long the operation should last before timing out within the range of 1 to 300 seconds.

When a loop is detected, a loop report is displayed in response to the DlswLoopDetect command. For example, if a loop is detected in the example on NETBuilder A a report similar to the following appears:

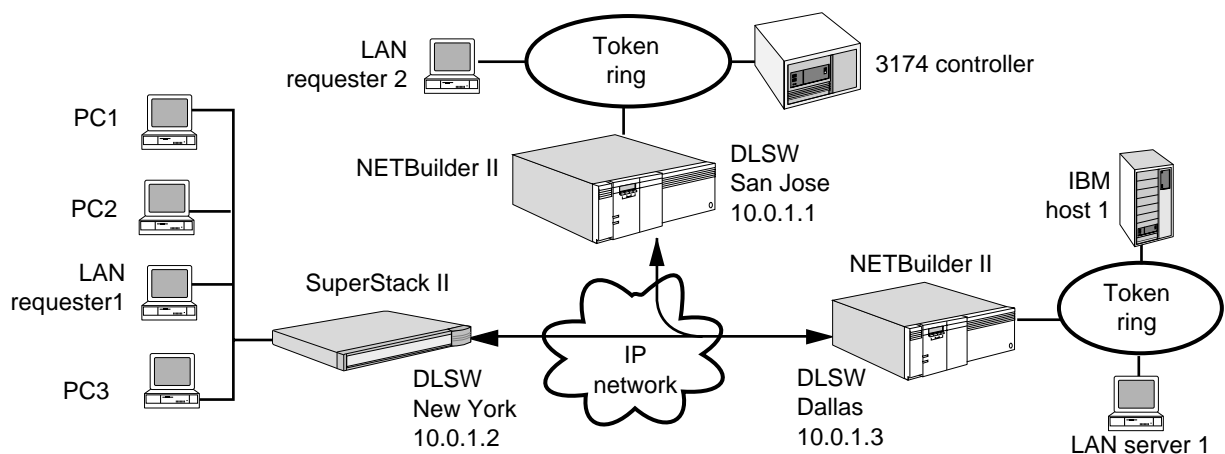
```
Loop detected:
  Originator: 129.213.1.11
  Received from: 129.213.1.11 at 129.213.2.22
  Received on port 1 at 129.213.2.33
  Received from: 129.213.2.33 at 129.213.1.44
  Received on port 1 at 129.213.1.11
End of loop report
```

How Data Link Switching Works

DLSw supports SNA and NetBIOS in multiprotocol routers. SNA and NetBIOS provide connection-oriented services. SNA and NetBIOS use IEEE 802.2 LLC2 protocol over LANs. DLSw also provides SNA connectivity over WAN links for devices attached by SDLC peripheral links. For conceptual information on how data link switching works for LANs, see RFC 1795. The NETBuilder bridge/router family of hardware and software fully implements this standard.

Figure 260 shows a typical network configuration using data link switching with SNA and NetBIOS traffic to connect three bridge/routers across an IP internetwork. Each connection is a tunnel, which consists of two TCP ports: one to send data (port #2067) and one to receive data (port #2065).

Figure 260 Simple Data Link Switching Configuration



Multiple sessions between different ports are multiplexed onto a single tunnel. For instance, if there is a session connecting LAN server 1 and LAN requester 1, and a concurrent session connecting PC1 and host 1, traffic is multiplexed onto a single tunnel between the NETBuilder II bridge/router at Dallas and the SuperStack II NETBuilder bridge/router at New York.

Media Addressing and NetBIOS Name Caching

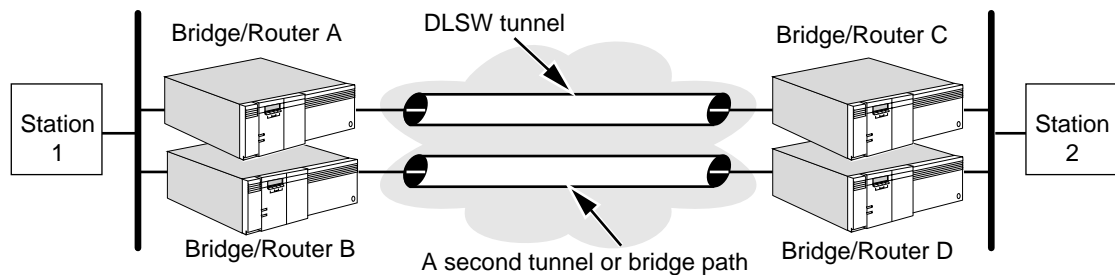
When the 3Com DLSw router receives an explorer or NetBIOS name type frame, the router first checks the statically defined table for the existence of a predefined route. The router also checks the DLSw caching tables for a match. If a match is found, the frame is forwarded on the static or cached DLSw tunnel. If no match exists, then the frame is forwarded to each DLSw tunnel. When the DLSw router receives a DLSw explorer or NetBIOS name type frame, the router adds the media address or the NetBIOS name to its caching tables.

A cached item is deleted when the DLSw router uses a cached route to forward an explorer frame but fails to get a response. The result is that the first explorer or query frame is sent using the cache tunnel. When that frame fails to get a response, the cached item is deleted and the query is resent on all tunnels.

DLSw Configuration and STP

DLSw is not aware of the Spanning Tree Protocol (STP). Because of this limitation, you must avoid configuring a second data path that can loop SNA and NetBIOS traffic back to an originating router. Do not configure either bridge or tunnel paths as second data paths. Avoid the topology shown in Figure 261 because it may duplicate packets and cause failure.

Figure 261 Illegal DLSw Tunneling Configuration



Your site may require redundancy in a DLSw environment. If you need DLSw bridge/router redundancy, contact your network supplier for planning.

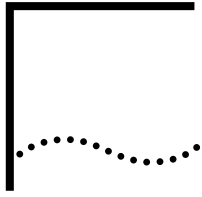


In token ring topologies, DLSW or LLC2 tunneling can support parallel paths in a source-routed-only environment.

Data Link Switching Terms

Data link switching terminology uses tunneling terms that have specific meanings defined in RFC 1795, and are relevant for the DLSw environment.

| | |
|---------------------------------|---|
| data link switching | A method for forwarding SNA and NetBIOS traffic between routers. |
| initial bandwidth | An option that allows you to define initial tunnel bandwidth. |
| instance ID | A number that identifies an entry in a table. |
| peer | A relationship between a local and remote router, usually referring to a remote router with a remote address, which is the peer IP address. |
| prioritization | An option that allows you to allocate tunnel bandwidth to data traffic coming from devices associated with a specific priority criterion. |
| Switch-to-Switch Protocol (SSP) | The protocol used between two communicating data link switches. |
| tunnel | A connection between two routers using two IP addresses, one in each router. Both routers must be using the data link switching Switch-to-Switch Protocol. Multiple tunnels between multiple routers can be configured. |
| tunnel ID | A local identifier that defines tunnels to peer devices. |
| tunneling | The encapsulation of SNA and NetBIOS traffic in a TCP/IP packet, using the Data Link Switching Protocol. |



CONFIGURING MULTICAST DATA LINK SWITCHING FOR NETBIOS AND SNA NETWORKS

This chapter describes how to configure your system to perform multicast data link switching (DLSw). Multicast DLSw allows easier scalability of large DLSw networks while reducing the number of configuration steps required. Multicast DLSw provides an enhancement to the RFC 1795-compliant DLSw described in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter. Multicast DLSw provides the following enhancements:

- Reduced configuration for data link switches
With RFC 1795-compliant DLSw, each data link switch in partially meshed or fully meshed networks must be configured for one or more peers so that TCP connections can be established between the DLSw peers. With multicast DLSw, IP multicast addresses are used for exploration, which eliminates the requirement that DLSw peers must be configured.
- Reduced WAN backbone traffic
With RFC 1795-compliant DLSw, each data link switch sends out broadcast CANUREACH_ex Switch-to-Switch Protocol (SSP) requests on every TCP connection. With multicast DLSw, only one multicast packet is sent out by a data link switch, which reduces WAN backbone traffic.
- Reduced TCP Overhead
With RFC 1795-compliant DLSw, each data link switch has two TCP connections with each of its peers, whether or not a circuit is established between end systems through the DLSw peers. With multicast DLSw, TCP connections are brought up only if a circuit needs to be established between the data link switches. The TCP connections are brought down when all circuits using the connection have ended.



Before you configure multicast DLSw, MOSPF must be configured. For information, see the Configuring IP Multicast Routing chapter and the MOSPF Service Parameters chapter in Reference for Enterprise OS Software.

Configuring Multicast DLSw

This section describes how to configure multicast DLSw for NetBIOS or SNA traffic. Multicast DLSw is useful in the following network topologies:

- Configurations in which stations are communicating using NetBIOS, where logical meshed network connectivity is desirable.
- SNA networks in which TCP connections between the PU2 client and the host server is always required. Multicast DLSw is useful in demand-based situations where the sessions between the clients and the host do not need to be up all the time. As a result, TCP connections do not need to be kept up all the time, saving processing overhead.

For configuration procedures for NetBIOS, see the next section. For configuration procedures for SNA client and server environments, see “Configuring Multicast DLSw for SNA Client and Server Environments” later in this chapter.



DVMRP is not supported with DLSw multicast.

Configuring DLSw Multicast for NetBIOS Mesh Environments

This section describes how to configure multicast DLSw for NetBIOS meshed environments. In this configuration, DLSw bridge/routers can use the default multicast address for both transmit and receive traffic.

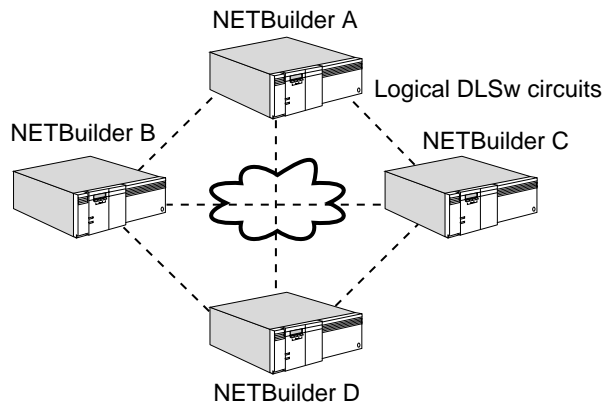
Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the IP addressing and IP routing protocols on the appropriate ports according to the Configuring IP Routing chapter.
- Configure MOSPF according to the procedures in the Configuring IP Multicast Routing chapter.
- Obtain the IP addresses for bridge/routers on each side of the TCP/IP connection.
- Configure the DLSw peer configuration according to the procedures described in “Configuring for NetBIOS” in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

Figure 262 is an example of a meshed NetBIOS environment. In the figure, four DLSw bridge/routers are participating in the multicast environment.

Figure 262 DLSw Multicast Example (Meshed NetBIOS Environment)



To configure multicast DLSw as shown in the figure, follow these steps on each bridge/router:

- 1 Set the DLSw mode on the bridge/router to multicast by entering:

```
SETDefault -DLSw MOde = Multicast
```



When configuring the bridge/router for multicast mode, you can also include the RouteInfo/NoRouteInfo parameter in the SETDefault command. This parameter

applies only when DLSw is configured to operate under multicast mode and specifies whether DLSw loop detection passes along route information when sending a loop frame to a multicast address. The default is NoRouteInfo. For more information, see “Enabling DLSw Loop Detection” in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

- 2 Enable NetBIOS for DLSw by entering:

```
SETDefault -DLSw CONTROL = EnableNetBios
```

- 3 Enable IP routing by entering:

```
SETDefault -IP CONTROL = Route
```

- 4 Enable multicast IP routing by entering:

```
SETDefault -MIP CONTROL = Enable
```

- 5 Enable OSPF on the DLSw port using:

```
SETDefault !<port> -OSPF CONTROL = Enable
```

- 6 Enable multicast OSPF on the DLSw WAN port using:

```
SETDefault !<port> -MOSPF CONTROL = Enable
```

After you follow these steps on each bridge/router, the routers send multicast requests onto the meshed network, and each router can reach every other router without configuring static DLSw peers.

Configuring Multicast DLSw for SNA Client and Server Environments

This section describes how to configure multicast DLSw for SNA client and server environments. In these configurations, one data link switch router is connected to an SNA host and a second data link switch router is connected to clients (PU2). You need to configure the appropriate multicast DLSw addresses on the client and the server routers.

Because you need to configure the multicast DLSw address on both sides, the benefit of using DLSw multicast for SNA client and server environments is limited. The primary benefit of using DLSw multicast instead of RFC 1795-compliant DLSw is that the TCP connections come up dynamically as needed and go down when the circuit becomes idle.

Prerequisites

Before beginning this procedure, complete these tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the IP addressing and IP routing protocols on the appropriate ports according to the Configuring IP Routing chapter.
- Configure MOSPF according to the procedures in the Configuring IP Multicast Routing chapter.
- Obtain the IP addresses for bridge/routers on each side of the TCP/IP connection.
- Configure the DLSw peer configuration according to the procedures described in “Configuring for SNA” in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

To configure multicast DLSw on the SNA client, follow these steps:

- 1 Set the DLSw mode on the bridge/router to multicast by entering:

```
SETDefault -DLSw MOde = Multicast
```

- 2 Delete the default multicast address by entering:

```
DELeTe -DLSw MulticastAddr DEFault
```

This command deletes the default multicast address 224.0.10.0, which allows you to configure the multicast address in the next step.

You can restore the default multicast address by entering the ADD -DLSw MulticastAddr command and specifying DEFault.

The default multicast address is configured as TxRx, which is acceptable for fully meshed configurations, but is not suitable for client-server configurations.

- 3 Define the Class D multicast address that the client bridge/router will *receive* SNA traffic on using:

```
ADD -DLSw MulticastAddr <IP multicast address> SNA Rx
```

When entering the IP multicast address, you can enter any Class D address, from 224.0.0.0 to 239.255.255.255. The range of valid multicast addresses for DLSw multicast only is from 224.0.10.0 to 224.0.10.255.

For example, to add the IP multicast address 224.0.10.100 to receive traffic on, enter:

```
ADD -DLSw MulticastAddr 224.0.10.100 SNA Rx
```

- 4 Define the Class D multicast address that the client bridge/router will *transmit* SNA traffic on using:

```
ADD -DLSw MulticastAddr <IP multicast address> SNA Tx
```

For example, to add the IP multicast address 224.0.10.200 to receive traffic on, enter:

```
ADD -DLSw MulticastAddr 224.0.10.200 SNA Tx
```

- 5 Enable IP routing by entering:

```
SETDefault -IP CONTrol = Route
```

- 6 Enable multicast IP routing by entering:

```
SETDefault -MIP CONTrol = Enable
```

- 7 Enable OSPF on the port using:

```
SETDefault !<port> -OSPF CONTrol = Enable
```

- 8 Enable multicast OSPF on the port using:

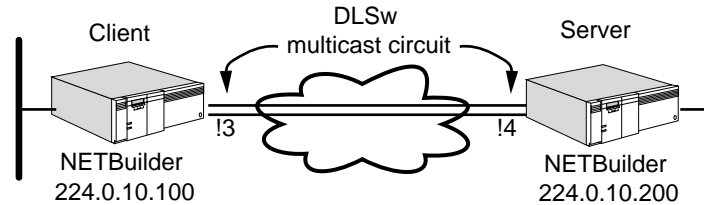
```
SETDefault !<port> -MOSPF CONTrol = Enable
```

To configure multicast DLSw on the SNA server, follow the steps in the previous procedure except for steps 3 and 4. In steps 3 and 4 configure the multicast addresses used to send and receive traffic, but reverse the addresses configured for those steps. On the server, configure the sending and receiving multicast addresses by entering:

```
ADD -DLSw MulticastAddr 224.0.10.200 SNA Rx
```

```
ADD -DLSw MulticastAddr 224.0.10.100 SNA Tx
```

Figure 263 is an example of an SNA configuration in which multicast DLSw is used. Table 64 lists the commands to configure on each DLSw client and server bridge/router to allow multicast DLSw to work.

Figure 263 DLSw Multicast Example (SNA Configuration)**Table 64** Commands to Configure Multicast DLSw for SNA

| Commands Entered on Client Bridge/Router | Commands Entered on Server Bridge/Router |
|---|---|
| SETDefault -DLSw MOde = Multicast | SETDefault -DLSw MOde = Multicast |
| DElete -DLSw MulticastAddr DEFault | DElete -DLSw MulticastAddr DEFault |
| ADD -DLSw MulticastAddr 224.0.10.100 SNA Rx | ADD -DLSw MulticastAddr 224.0.10.200 SNA Rx |
| ADD -DLSw MulticastAddr 224.0.10.200 SNA Tx | ADD -DLSw MulticastAddr 224.0.10.100 SNA Tx |
| SETDefault -IP CONTrol = Route | SETDefault -IP CONTrol = Route |
| SETDefault -MIP CONTrol = Enable | SETDefault -MIP CONTrol = Enable |
| SETDefault !3 -OSPF CONTrol = Enable | SETDefault !4 -OSPF CONTrol = Enable |
| SETDefault !3 -MOSPF CONTrol = Enable | SETDefault !4 -MOSPF CONTrol = Enable |

Customizing the DLSw Multicast Configuration

This section describes how to customize the multicast DLSw configuration.

Tuning DLSw Multicast Parameters

You can tune the retry interval and retry count for the number of times that the SSP frames sent on multicast are retried. The default retry interval is 3 seconds for SNA and 1 second for NetBIOS, and the default retry count is 0 (no retries). To change the retry interval and retry count, use:

```
SETDefault -DLSw McastRetry = <SNA | NetBios> <retry interval (1-5)>
<retry count (0-5)>
```

You must also specify whether the change is for SNA or NetBIOS traffic.

You can also specify the number of minutes that a TCP connection between multicast DLSw peers will stay up without any circuit using the connection. To change the TCP idle time, use:

```
SETDefault -DLSw McastTcpIdle = <timer duration (1-255)>
```

The default is 3 minutes.

Restoring the Default Multicast Address

If you want to restore the default multicast address (224.0.10.0) on the bridge/router after previously configuring an address for multicasting purposes, enter:

```
ADD -DLSw MulticastAddr DEFault
```

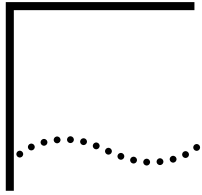
When you specify DEFault, the traffic type defaults to ALL and the usage defaults to TxRx.

**Disabling DLSw
Multicast**

To disable DLSw multicast on the bridge/router, enter:

```
SETDefault -DLSw MOde = NoMulticast
```

The bridge/router stops sending out multicasts to DLSw stations on the network.



CONFIGURING FRAME RELAY ACCESS DEVICE SUPPORT FOR SNA

This chapter describes how to configure the bridge/router as a Frame Relay Access Device (FRAD) node to provide Frame Relay access support for Systems Network Architecture (SNA). The FRAD functionality implements Logical Link Control, type 2 (LLC2) encapsulation over Frame Relay, and uses data packet encapsulation methods based on the RFC 1490 frame format. In SNA environments, FRAD provides similar functionality to the IBM Frame Relay Boundary Network Node (BNN) and Boundary Access Node (BAN).



For more information about how FRAD works, see “How the Frame Relay Access Device Works” later in this chapter.

Configuring the NETBuilder as a FRAD Node

This section describes how to configure the bridge/router as a FRAD node for both BAN and BNN configurations. Table 65 shows how BAN and BNN support FRAD nodes.

Table 65 BAN and BNN Capabilities

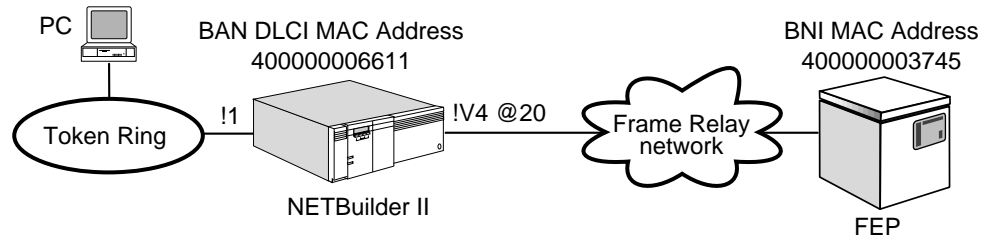
| BAN | BNN |
|--|--|
| RFC 1490 bridged token ring format. | RFC 1490 routed SNA format. |
| Static addressing not needed because the end-station MAC address is provided in every frame. | Static addressing using end-station MAC and SAP address. |
| Load balancing. | No load balancing. |
| Supports LAN, SDLC, and APPN. | LAN, SDLC, and APPN. |

Configuring FRAD for LAN-Attached End Stations

You can configure FRAD for LAN-attached end stations for either Boundary Access Node (BAN) or for Boundary Network Node (BNN). This section is divided into two procedures, one for BAN-attached end stations and one for LAN-attached end stations using BNN.

Configuring the FRAD Node for a BAN-Attached End Station

Figure 264 is an example of a NETBuilder II bridge/router acting as a FRAD node with a BAN-attached end station. The FRAD provides Frame Relay access to the remote host front-end processor (FEP).

Figure 264 FRAD Node Configuration (BAN-attached End Station)

Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your bridge/router according to the procedures described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- To control which BAN ports are active at a given time, you must configure the ports accessing Frame Relay as virtual ports.
- Configure the Frame Relay interface. For information on configuring Frame Relay, see the Configuring Wide Area Networking Using Frame Relay chapter.

To configure the NETBuilder bridge/router as a FRAD node for a BAN-attached end station, follow these steps:

- 1 Configure the physical Frame Relay port by entering:

```
SETDefault !4 -PATH BAud = 56
SETDefault !4 -PORT OWNeR = FrameRelay
```

- 2 Add the virtual port by entering:

```
ADD !v1 -PORT VirtualPort 4@40
```

- 3 Configure the address mapping for Frame Relay connections to the FEP using:

```
ADD !<port> -DLsw BoundaryAccessNode <ban dlcI mac addr> [<bni mac addr>]
```

With this syntax you map the source MAC to the FEP MAC and assign the boundary node indicator (BNI) MAC address. For example:

```
ADD !v1 -DLsw BAN 400000006611 400000003745
```

For more information about the mapping rules that apply to the `FradMap` parameter, see “How the Frame Relay Access Device Works” later in this chapter. For more information about the `FradMap` parameter, see the `DLSw Service Parameters` chapter in *Reference for Enterprise OS Software*.

- 4 Enable LLC2 control on the Frame Relay port by entering:

```
SETDefault !v1 -LLC2 CONTrol = Enabled
```

- 5 Enable LLC2 control on the local port to enable host-initiated activation by entering:

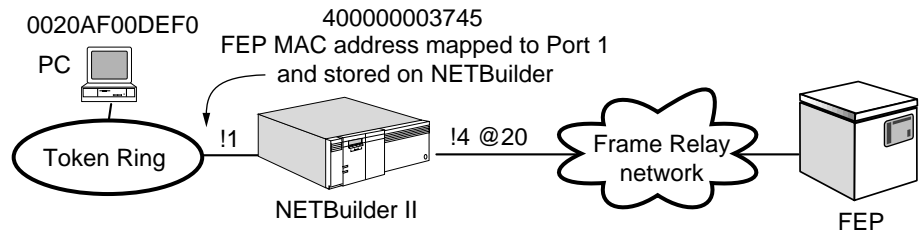
```
SETDefault !1 -LLC2 CONTrol = Enabled
```

- 6 For the SuperStack II NETBuilder Token Ring platforms or source-routing-only environment, you need to configure the end station support for LLC2 and source routing for the LAN port.

Configuring the FRAD Node for a LAN-Attached End Station Using BNN

Figure 265 is an example of a NETBuilder II bridge/router acting as a FRAD node with a LAN-attached end station. The FRAD provides Frame Relay access to the remote host FEP.

Figure 265 FRAD Node Configuration (LAN-attached End Station using BNN)



Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your bridge/router according to the procedures described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the Frame Relay interface. For information, see the Configuring Wide Area Networking Using Frame Relay chapter.

To configure the NETBuilder bridge/router as a FRAD node for a LAN-attached end station using BNN, follow these steps:

- 1 Configure the physical Frame Relay port by entering:

```
SETDefault !4 -PATH BAud = 56
SETDefault !4 -PORT OWNeR = FrameRelay
```

The baud rate for the path should match the speed of the Frame Relay line.

- 2 For LAN-attached end stations using BNN, disable bridging on the Frame Relay port by entering:

```
SETDefault !4 -BR TransparentBRidge = NoTransparentBRidge
SETDefault !4 -SR SrcRouBRidge= NoSrcRouBRidge
```

- 3 To configure the address mapping for the Frame Relay connections to the FEP use:

```
ADD !<port> -DLSw FradMap <src mac> <src sap> <fep mac> <fep sap> <DLCI>
<code point>
```

With this syntax, you map the source MAC and SAP to the FEP MAC and SAP and assign the DLCI and the code point. For example:

```
ADD !4 -DLSw FradMap 0020AF00DEF0 4 400000003745 4 20 82
```

For more information about the mapping rules that apply to the FradMap parameter, see "How the Frame Relay Access Device Works" later in this chapter. For more information about the FradMap parameter, see the DLSw Service Parameters chapter in *Reference for Enterprise OS Software*.

- 4 Enable LLC2 control on the Frame Relay port by entering:

```
SETDefault !4 -LLC2 CONTrol = Enabled
```

- 5 To enable LLC2 control on the local port to enable host-initiated activation enter

```
SETDefault !1 -LLC2 CONTrol = Enabled
```

- For the SuperStack II NETBuilder Token Ring bridge/routers or in a source-routing-only environment, you need to configure the end station support for LLC2 and source routing for the LAN port

Configuring FRAD for SDLC-Attached End Stations

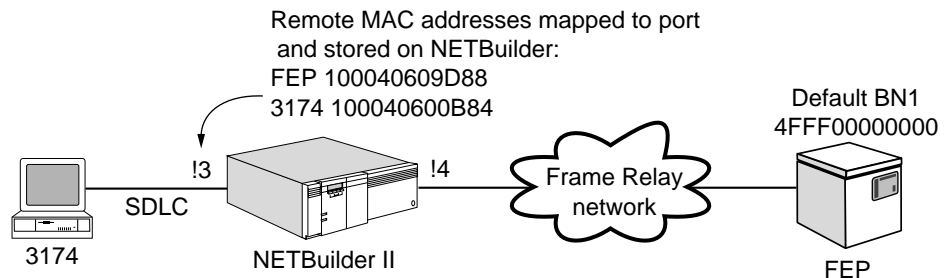
You can configure FRAD for SDLC-attached end stations for either Boundary Access Node (BAN) or for Boundary Network Node (BNN). This section is divided into two procedures, one for BAN and one for BNN.

Configuring the FRAD Node for an SDLC-Attached End Station Using BAN

Figure 266 is an example of a NETBuilder II bridge/router acting as a FRAD node with an SDLC-attached end station. The addressing shown is for a BNN configuration.

For a BAN configuration, the FEP address 100040609D88 is seen internally at the NETBuilder II (FRAD node) bridge/router and is mapped to the BAN BNI MAC address 4FFF00000000. The FRAD provides Frame Relay access to the remote host FEP.

Figure 266 FRAD Node Configuration (SDLC-attached End Station) for BAN



Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your bridge/router according to the procedures described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the Frame Relay interface. For information, see the Configuring Wide Area Networking Using Frame Relay chapter.
- Configure the SDLC conversion for the SDLC-attached end station. This configuration must be done before configuring the NETBuilder II bridge/router as a FRAD node. For more information, see the Configuring SDLC and HDLC Tunneling for SNA Networks chapter.

Procedure

To configure the NETBuilder II bridge/router as a FRAD node for an SDLC-attached end station for BAN, follow these steps:

- To configure the SDLC physical port and path attributes enter:

```
SETDefault !3 -PORT OWNeR = SDLC
SETDefault !3 -PATH DUplex = Full
SETDefault !3 -PATH ENCOding = NRZI
```

- To configure the SDLC logical port and path attributes enter:

```
ADD !3 -SDLC PortCU PU31741
```

```
SETDefault !3 -SDLC PDatMode = Full
SETDefault !3 -SDLC PROle = Primary
SETDefault !3 -SDLC PCONTrol = Enabled
```

- 3 To configure the SDLC CU (adjacent link station) attributes enter.

```
SETDefault !PU31741 -SDLC CUAddr = C1
SETDefault !PU31741 -SDLC CULocalMac = 100040600B84
SETDefault !PU31741 -SDLC CUREmoteMac = 100040609D88
SETDefault !PU31741 -SDLC CULocalSap = 4
SETDefault !PU31741 -SDLC CUREmoteSap = 4
SETDefault !PU31741 -SDLC CUCONTROL = Enabled
```

- 4 To configure address mapping for the Frame Relay connections to the FEP use.

```
ADD !<port> -DLSw BoundaryAccessNode <ban dlci mac addr> [<bni mac addr>]
```

With this syntax, you map the source MAC to the FEP MAC and assign the BNI MAC address. For example, assuming that the DLCI is 40:

```
ADD !v4 -PORT VirtualPort 4@40
ADD !v1 -DLSw BAN 100040600D88
```

If the BNI DLCI address is different, you must add the BNI DLCI address to the command. For example:

```
ADD !v1 -DLSw BAN 100040600D88 4FFF00037451
```

For more information about the mapping rules that apply to the FradMap parameter, see “How the Frame Relay Access Device Works” later in this chapter. For more information about the FradMap parameter, see the DLSw Service Parameters chapter in *Reference for Enterprise OS Software*.

- 5 Enable LLC2 control on the Frame Relay port by entering

```
SETDefault !4 -LLC2 CONTROL = Enabled
```

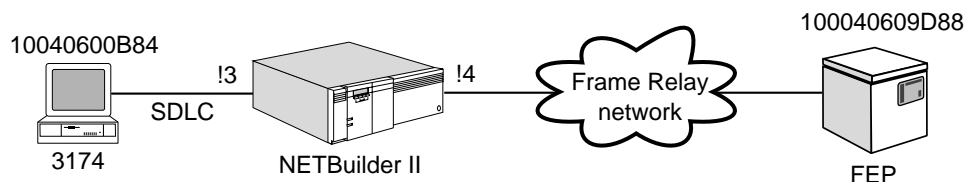
- 6 Enable LLC2 control on the local port to enable host-initiated activation by entering:

```
SETDefault !1 -LLC2 CONTROL = Enabled
```

Configuring the FRAD Node for an SDLC-Attached End Station Using BNN

Figure 267 shows an example of a NETBuilder II bridge/router acting as a FRAD node with an SDLC-attached end station for BNN.

Figure 267 FRAD Node Configuration (SDLC-attached End Station) for BNN



Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your bridge/router according to the procedures described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

- Configure the Frame Relay interface. For information, see the Configuring Wide Area Networking Using Frame Relay chapter.
- Configure the SDLC conversion for the SDLC-attached end station. This configuration must be done before configuring the NETBuilder II bridge/router as a FRAD node. For more information, see the Configuring SDLC and HDLC Tunneling for SNA Networks chapter.

Procedure

To configure the NETBuilder II bridge/router as a FRAD node for an SDLC-attached end station for BNN, follow these steps:

- 1 Configure the physical Frame Relay port by entering:

```
SETDefault !4 -PATH BAud = 56
SETDefault !4 -PORT OWNer = FrameRelay
```

The baud rate for the path should match the speed of the Frame Relay line.

- 2 Configure the SDLC physical port and path attributes by entering:

```
SETDefault !3 -PORT OWNer = SDLC
SETDefault !3 -PATH DUplex = Full
SETDefault !3 -PATH ENCOding = NRZI
```

- 3 Configure the SDLC logical port and path attributes by entering:

```
ADD !3 -SDLC PortCU PU31741
SETDefault !3 -SDLC PDatMode = Full
SETDefault !3 -SDLC PROle = Primary
SETDefault !3 -SDLC PCONTrol = Enabled
```

- 4 Configure the SDLC CU (adjacent link station) attributes by entering:

```
SETDefault !PU31741 -SDLC CUAddr = C1
SETDefault !PU31741 -SDLC CULocalMac = 100040600B84
SETDefault !PU31741 -SDLC CUREmoteMac = 100040609D88
SETDefault !PU31741 -SDLC CULocalSap = 4
SETDefault !PU31741 -SDLC CUREmoteSap = 4
SETDefault !PU31741 -SDLC CUCONTrol = Enabled
```

- 5 For BNN-attached end stations, disable bridging on the Frame Relay port by entering:

```
SETDefault !4 -BR TransparentBRidge = NoTransparentBRidge
SETDefault !4 -SR SrcRouBRidge = NoSrcRouBRidge
```

- 6 Configure the address mapping for Frame Relay connections to the FEP using:

```
ADD !<port> -DLsw FradMap <src mac> <src sap> <fep mac> <fep sap> <DLCI> <code point>
```

With this syntax, you map the source MAC and SAP to the FEP MAC and SAP and assign the DLCI and the code point. For example:

```
ADD !4 -DLsw FradMap 100040600B84 4 100040609D88 4 20 82
```

For more information about the mapping rules that apply to the FradMap parameter, see “How the Frame Relay Access Device Works” later in this chapter. For more information about the FradMap parameter, see the DLsw Service Parameters chapter in *Reference for Enterprise OS Software*.

- 7 Enable LLC2 control on the Frame Relay port by entering:

```
SETDefault !4 -LLC2 CONTrol = Enabled
```

- 8 Enable the LLC2 control on the local port to enable host-initiated activation by entering:


```
SETDefault !1 -LLC2 CONTROL = Enabled
```

Deleting Frame Relay Address Mappings

To delete a Frame Relay address mapping for BAN, use:

```
DElete !<vport> -DLSW BoundaryAccessNode <bnn dlci mac addr>
```

For example, to delete an address mapping on virtual port 1 with a FEP MAC address of 400000006611, enter:

```
DElete !v2 -DLSW BoundaryAccessNode 400000006611
```

To delete a Frame Relay address mapping for BNN use:

```
DElete !<port> -DLSw FradMap <src mac> <src sap> <fep mac> <DLCI> <code point>
```

For example, to delete an address mapping on port 2 with a source MAC address of 00608C2C61B5, a source SAP of 04, and a FEP MAC address of 40005A65BED, enter:

```
DElete !2 -DLSw FradMap 00608C2C61B5 04 40005A65BED
```

Displaying Frame Relay Address Mappings

To display Frame Relay address mappings, use:

```
SHow [!<port>|!*] -DLSw FradMap
```

How the Frame Relay Access Device Works

The 3Com FRAD implements LLC2 encapsulation over Frame Relay based on the RFC 1490 frame format. RFC 1490 describes how FRAD carries LLC2 frames over Frame Relay.

The specific values of SNA and NetBIOS are documented in the Frame Relay Forum *FRF.3 Multiprotocol Encapsulation Implementation Agreement* document, as well as in the ANSI *T1.617 Annex F*. These values are referred to as *code points* in these documents. The following are the code points currently defined for SNA and NetBIOS:

0x81- SNA Subarea (FID4)

0x82 - SNA Peripheral (FID2)

0x83 - SNA APPN (FID2)

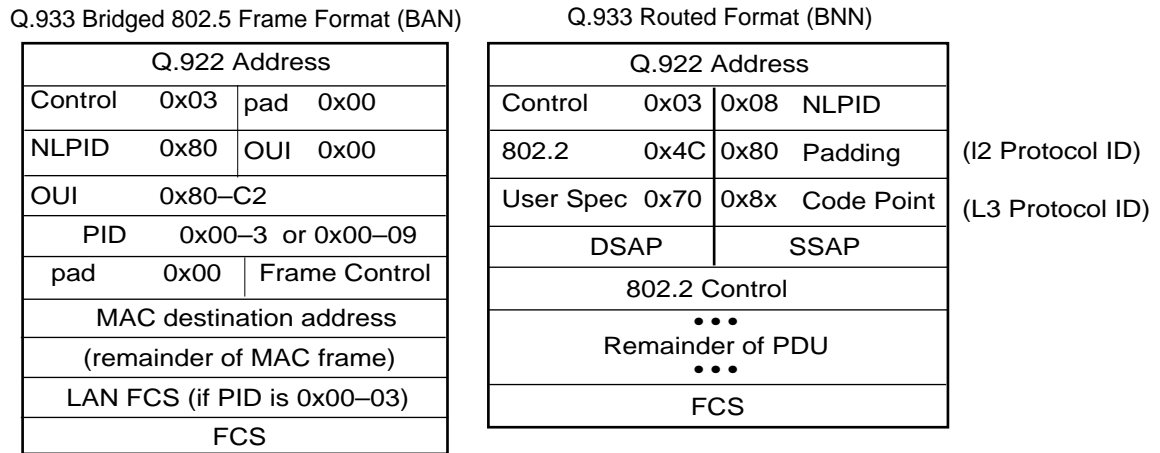
0x84 - NetBIOS

These code points are used only by BNN for routed frames. BAN uses MAC address mapping and does not require code points.

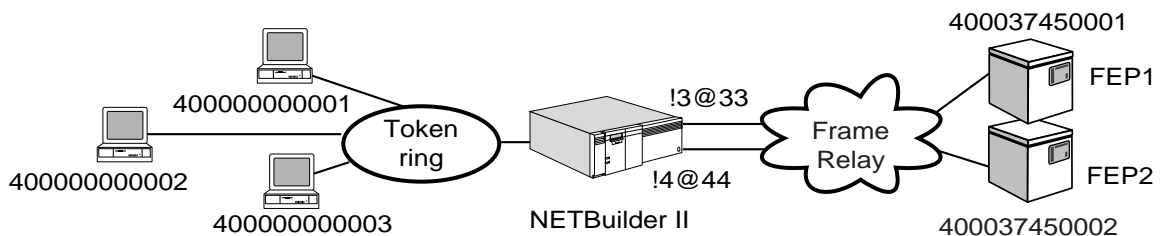


The 3Com FRAD implementation does not support 0x81 and 0x84 data traffic. For 0x82, NCP processes the data as non-APPN peripheral data. For 0x83, NCP processes the data as APPN peripheral data.

Figure 268 shows the RFC 1490 encapsulation format for SNA and NetBIOS.

Figure 268 RFC 1490 Encapsulation Format for BAN and BNN Implementation

BNN Configuration RFC 1490 encapsulation eliminates the bridge/router on the host side; the SNA over the Frame Relay connection is terminated directly by the FEP. Figure 269 shows a configuration in which the Frame Relay connection is terminated directly to the FEPs in a BNN configuration.

Figure 269 SNA Over Frame Relay Terminated by FEPs BNN Implementation

RFC 1490 encapsulation is implemented in NCP V7R1. With this implementation, the FEP can be attached directly to the Frame Relay network. This reduces the number of conversion points in the network, and can result in fewer network failure points for SNA traffic. BAN uses NCP V7R3. Updates are available for NCP V7R1 and V7R2.

Another advantage of RFC 1490 encapsulation is that it carries only LLC2 information (without the bridging frames), which places less overhead on the network because no broadcasting occurs (that is, no fan-out of frames to other bridging ports in the bridge/router). By eliminating bridging and broadcast frames, performance can be improved.

Since no broadcasting occurs, the NETBuilder bridge/router must specifically map the incoming MAC and SAP address to a specific outbound datalink connection identifier (DLCI), and vice versa. This mapping is performed using the -DLSw FradMap parameter syntax as follows:

```
ADD !<port> -DLSw FradMap <src mac> <src sap> <fep mac> <fep sap>
    <DLCI> <code point>
```

In the syntax, <src mac> and <src sap> are the MAC and SAP addresses of the SNA end station.

For RFC 1490 encapsulation, the <fep sap> value, in conjunction with the <src sap> and <code point> values, is required. For outbound (host bound) traffic, the value specified for <fep sap> will be placed in the source service access point (SSAP) field, while the value specified for <src sap> will be placed in the destination service access point (DSAP) field (see Figure 268). The value specified for <fep sap> must match the address specified in the DLCADDR keyword on the VTAM PATH definition statement; for more information, see the *IBM VTAM Resource Definition Reference* and the *NCP/SSP/EP Resource Definition Guide*.

The <src mac> and <fep mac> values must be specified in noncanonical format. The <src sap> and <fep sap> values must be in the range of 0-FC and divisible by 4.

For outbound (host bound) traffic, the bridge/router uses <src mac>, <src sap>, and <fep mac> to find the mapped Frame Relay partner in the mapping table. For inbound traffic (from the host), the bridge/router uses the combination of <fr port>, <dldci>, and <fep sap> to find the mapped LLC2 partner in the mapping table.



The maximum number of mapping entries allowed in the mapping table is 250.

When configuring the bridge/router for FRAD, follow these mapping rules:

- The combination of <src mac>, <src sap>, and <fep mac> in the mapping table must be unique.
- The combination of <fr port>, <fep sap>, and <dldci> in the mapping table must be unique.
- The mapping between (<src mac>, <src sap>, <fep mac>) and (<FR port>, <fep sap>, <dldci>) must be one-to-one.

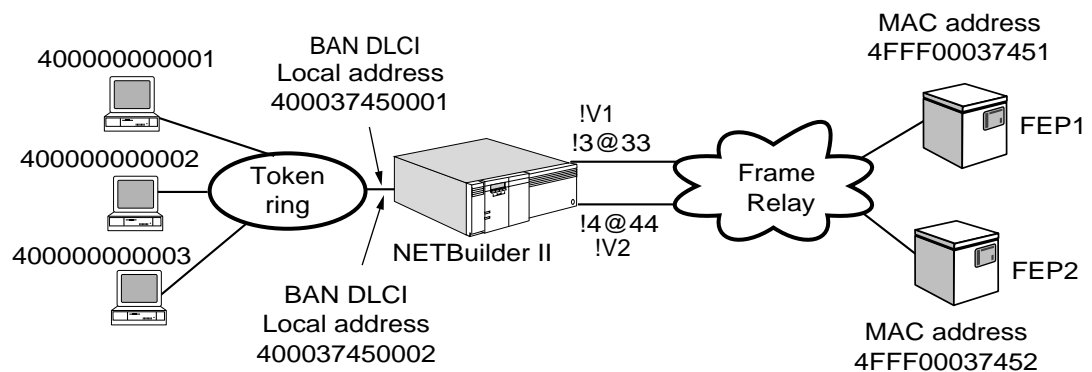
By following these mapping rules, you can multiplex more than one SNA Link connection on a single DLCI.

The FRAD uses the local switching feature of DLSw, and inherits the advantages of DLSw such as local termination of data link traffic. However, the FRAD is also subject to the same limitations as DLSw. For example, FEP-to-FEP (FID4) traffic is not supported. For more information on DLSw, see the Configuring Multicast Data Link Switching for NetBIOS and SNA Networks chapter.

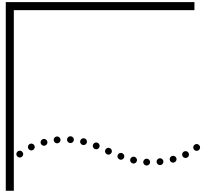
BAN Configuration BAN uses the RFC 1490 encapsulation implemented in NCP V7R3. BAN works similarly to BNN. The difference is that BNN uses setup coding to recognize each FEP and BAN uses MAC addressing.

Figure 270 shows a configuration in which the Frame Relay connection is terminated directly to the FEPs in a BAN configuration.

Figure 270 SNA Over Frame Relay Terminated by FEPs, BAN Implementation



Each Frame Relay BAN PVC is assigned a MAC address that is known as the BAN DLCI MAC address. This MAC address is assigned a virtual port and the BAN device listens for this MAC address on the LAN ports. The FEP recognizes the MAC address as a BNI. Both the BAN DLCI MAC and the BNI address may be the same. If they are different, the BAN device maps the two addresses. A test frame is sent to the BAN DLCI address. The first device that answers is the device to which the connection is made. This connection occurs when LLC2 receives the frames for a LAN device being sent to the BAN DLCI MAC address. Since no device is registered for the address, the frame is then forwarded to datalink switching (DLSw), which forwards the frame to all its tunnels including the one for local switching. Since the frame was received on a LAN port, it will not be forwarded out of any LAN port.



CONFIGURING LAN ADDRESS ADMINISTRATION

This chapter describes how to use LAN Address Administration (LAA) to assign a media access control (MAC) address to a physical path or to the main processor module interface, overriding the MAC address burned in the PROM on the physical interface.



Assigning MAC addresses to a path or main processor interface is supported for token ring, Ethernet, and FDDI ports only.

By assigning a MAC address to a path, you can use the same MAC address for multiple paths for load splitting purposes in Systems Network Architecture (SNA) environments. By assigning a MAC address to a path that is different from the MAC address burned in the physical interface PROM, you can hotswap modules on a port and still maintain the same MAC address.



CAUTION: *Using LAA on paths being used to route DECnet network traffic can cause problems in DECnet environments. For more information, see “Using LAA with DECnet” later in this chapter.*



LAA cannot coexist with the IP VRRP feature on the same bridge/router.

Assigning a MAC Address to a Physical Path

To assign a MAC address to a physical path, follow these steps:

- 1 To assign a MAC address to a physical path (a different address from the one burned on the module PROM), use:

```
SETDefault !<path> -PATH MacAddress = %<MAC address> | Mac <MAC address> |  
Ncmac <MAC address>
```

You can enter the MAC address in one of two formats: canonical or noncanonical. To enter the address in canonical format, precede the address with the prefix “Mac” or the percent symbol (%). To enter the address in noncanonical format, precede the address with the prefix Ncmac. If you precede the MAC address with Mac or Ncmac, enter a space between the prefix and the address. If you precede the MAC address with the percent symbol (%), do not enter a space between the symbol and the address.

Bits 0 and 1 of the first byte must be set to 0 and 1 respectively. Bit 1 is the universally and locally administered bit. This limits the choice of addresses to the following set (where x can have any value).

In canonical format:

```
x2xx xxxx xxxx  
x6xx xxxx xxxx  
xAxx xxxx xxxx  
xExx xxxx xxxx
```

In noncanonical format:

```
4xxx xxxx xxxx
5xxx xxxx xxxx
6xxx xxxx xxxx
7xxx xxxx xxxx
```

For example, to assign the canonical address 020002033D76 to path 2, enter:

```
SETDefault !2 -PATH MacAddress = Mac 020002033D76
```

To assign the noncanonical address 400040C0BC6E to path 2, enter:

```
SETDefault !2 -PATH MacAddress = NcMac 400040C0BC6E
```

To assign the MAC address 020002030EF2 in canonical format for token ring, enter:

```
SETDefault !2 -PATH MacAddress = %020002030EF2
```

To convert a MAC address from canonical format to noncanonical format and vice-versa, use the MacAddrConvert command. For more information, see the Commands chapter in *Reference for Enterprise OS Software*.



CAUTION: Do not assign a multicast address for the MAC address. Also, do not assign a MAC address that is either a smart filtering MAC address or one of the bridge BPDU addresses.

- 2 After you have reassigned the MAC address, re-enable the path using:

```
SETDefault !<path> -PATH CONTROL = Enabled
```

After you re-enable the path, the new MAC address assigned to the path will be shown when MAC addresses for any protocol are displayed. The new address remains assigned to the interface until you specifically reset the address. The new address remains assigned after you reboot the bridge/router.

- 3 If the LAA address is used by Advanced Peer-to-Peer Networking (APPN), you must deactivate and then activate the node control for the new address to be effective by entering:

```
SETDefault -APPN CONTROL = Deactivate
```

```
SETDefault -APPN CONTROL = Activate
```

- 4 Verify that the new MAC address has been assigned by entering:

```
SHow !* -PATH MacAddress
```

After you have assigned a MAC address to a path, you can reassign the path back to the MAC address burned on the PROM using:

```
SETDefault !<path> -PATH MacAddress = Reset
```

When you reset the MAC address, the address you previously assigned is deleted.

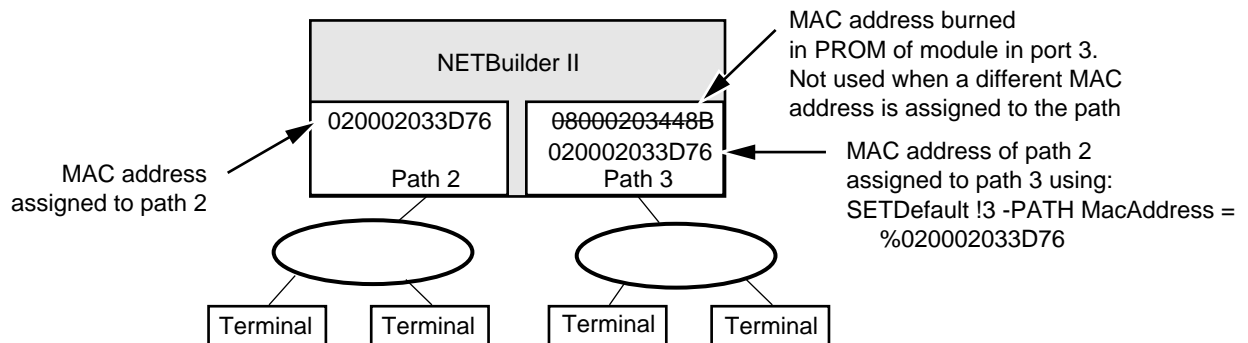
Figure 271 shows how you can use duplicate MAC addresses by reassigning an existing address. In the figure, you reassign the MAC address on path 3 to duplicate the address on path 2. The MAC address burned in the PROM of the module on path 3 still exists, but is not used for any connections. The MAC address burned in the PROM is transparent until the MAC address is reset.

You cannot set duplicate MAC addresses on the same ring in token ring environments. If you set duplicate MAC addresses on the same bridge/router, each path must be connected to different rings.



CAUTION: Setting duplicate MAC addresses is recommended only for SNA and other connection-oriented protocols. In addition, setting duplicate MAC addresses will work only in source routing LAN environments. As a result, setting duplicate MAC addresses is not recommended on transparent bridges or source route transparent bridges.

Figure 271 Setting Duplicate MAC Addresses Using LAA



Assigning a MAC Address to a Main Processor Module Interface

You can assign a MAC address to the main processor module interface on a NETBuilder II bridge/router. This can be useful in assigning a bridge/router to act as a backup network node in APPN environments.

To assign a MAC address to the main processor module interface on a NETBuilder II bridge/router, follow the procedures in "Assigning a MAC Address to a Physical Path" earlier in this chapter. However, when setting the MacAddress parameter, instead of specifying a path number, specify !0 to represent the processor module interface. For example, to assign the noncanonical address 400040C0BC6E to the interface, enter:

```
SETDefault !0 -PATH MacAddress = NcMac 400040C0BC6E
```

After you change the MAC address of the interface, you must reboot the bridge/router.

Using Duplicate MAC Addresses for SNA Load Balancing

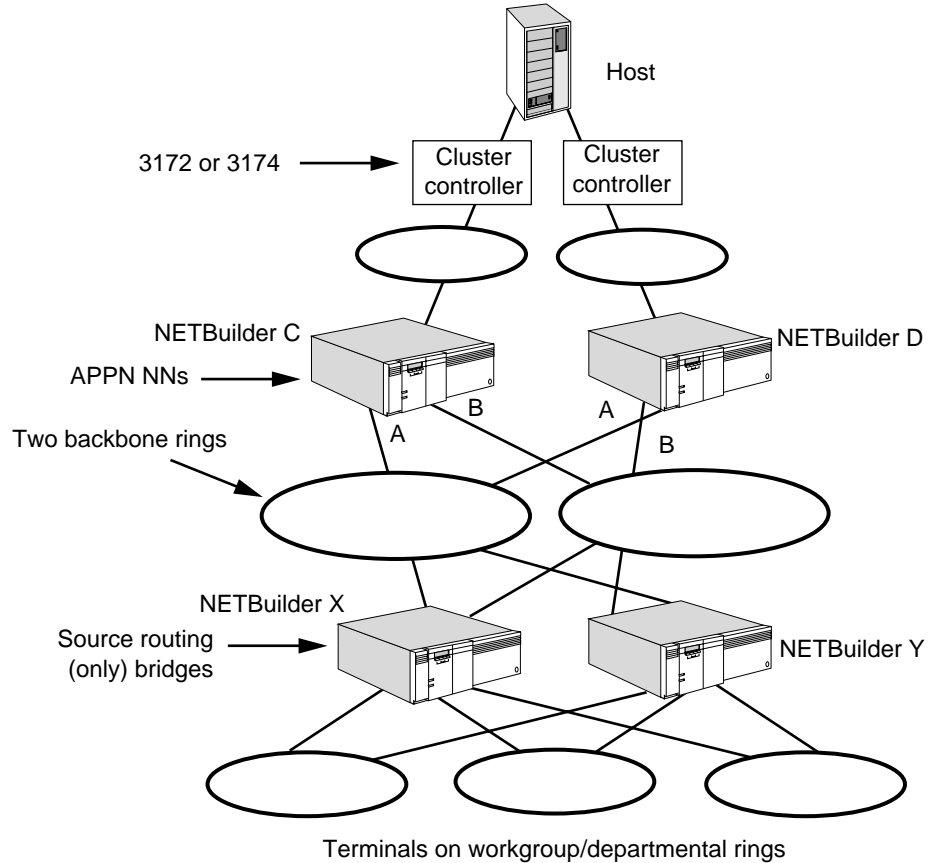
You can set duplicate MAC addresses to set up load balancing for SNA environments. In Figure 272, NETBuilder C and NETBuilder D are APPN network nodes in which duplicate MAC addresses are used on both so that connections to the host from the terminals at the bottom of the figure can go through either bridge/router.

In this example, MAC addresses A and B are duplicated on NETBuilder C and NETBuilder D. In this environment, the end stations at the bottom of the figure must configure the address of the host. Half of the end stations can configure MAC address A as the address to the host, and the other end stations can configure MAC address B. Two backbone rings are required because you cannot have two stations with the same MAC address on the same ring.

Using source routing, the end stations send out a discovery packet for the host address (either address A or B). The discovery packets are bridged through both bridges (NETBuilder X and NETBuilder Y) to the backbone rings. The discovery packet flows on both backbone rings. NETBuilder C and D both respond. The

workstation chooses the first path to respond. When the traffic on both bridges and rings is “load balanced,” if one bridge or ring goes down, the end stations can rediscover a new path to the host without reconfiguring.

Figure 272 Using LAA for SNA Load Balancing



Using LAA with DECnet

Because both LAA and DECnet involve overwriting MAC addresses, you must be careful that any changed MAC addresses are not overwritten when you configure LAA and DECnet together. Depending on whether you configure LAA or DECnet first, you can overwrite a previously configured address. The difference in the results is as follows:

- If you configure LAA first on a path and then enable DECnet over that same path, the MAC address you configured using LAA will be overwritten by the DECnet address.
- If you enable DECnet first on a path and then try to reassign the MAC address of that path using LAA, you will be unable to reassign the MAC address because DECnet will not allow it.

If the paths go down, that may also affect which MAC address is being used.

For example, if LAA is configured first on path 4 and then DECnet is enabled over that same path, the following sequence of events may take place:

- 1 You reassign the MAC address on path 4 through LAA by entering:

```
SETDefault !4 -PATH MacAddress
```

- 2 If you or the configuration file then enables DECnet routing over path 4 by entering:

```
SETDefault !4 -DECnet CONTROL = ROute
```

the MAC address configured in the previous step is overwritten.

- 3 If path 4 goes down and comes back up, it still has the DECnet-configured address.

- 4 If you then disable DECnet by entering:

```
SETDefault !4 -DECnet CONTROL = NoRoute
```

the MAC address of path 4 defaults to the address burned on the adapter's PROM.

- 5 If path 4 goes down again and comes back up, the MAC address used is the address reassigned using LAA.

If DECnet is enabled first on path 4 and you then attempt to reassign the MAC address using LAA, the following sequence of events takes place:

- 1 You or the configuration file enables DECnet routing on the path by entering:

```
SETDefault !4 -DECnet CONTROL = ROute
```

- 2 You attempt to reassign the MAC address through LAA by entering:

```
SETDefault !4 -PATH MacAddress
```

Since DECnet is enabled, you are prevented from doing so, and you receive a warning message. The path continues to use the address configured through DECnet.

- 3 If path 4 goes down and then comes back up, the path still uses the DECnet-configured MAC address.

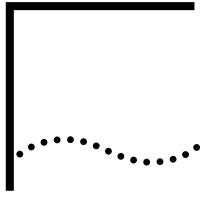
- 4 If you then disable DECnet by entering:

```
SETDefault !4 -DECnet CONTROL = NoRoute
```

the path now uses the MAC address burned in on the adapter's PROM.

- 5 If path 4 then goes down again and comes back up, the path continues to use the MAC address burned in on the adapter PROM.





CONFIGURING NETVIEW SERVICE POINT

This chapter describes how to configure a SSCP-PU session to a VTAM host. This feature supports SSCP-PU sessions to a VTAM host through PU4, PU5, and DLUr devices. This chapter also describes how to configure dependent LUs for BSC conversion.

Configuring NetView Service Point

To configure NetView Service Point for SSCP-PU session support on the bridge/router, follow these steps:

- 1 Configure the local node name and node ID using:

```
SETDefault -SNA LocalNodeName <netid.cpname> <node_id>
```

The local node ID is the ID block (ID BLK) followed by the ID number (ID NUM). Although the local node can communicate with multiple SSCPs, you can only have one node ID for the local node. The ID BLK and ID NUM is an eight-digit value and, and the values must match those configured on the VTAM host. The default ID BLK is 05D. Because the netid value must match the VTAM host configuration, obtain netid from your systems programmer.

For example, to configure the local node name US3COMHQ.NB2SF020 and node ID 01724001 (the ID BLK is 017 and the ID NUM is 24001), enter:

```
SETDefault -SNA LocalNodeName US3COMHQ.NB2SF020 01724001
```

- 2 Define the SNA port definition using:

```
SETDefault !<port> -SNA PortDef = <DLC type> (LLC2|FR|PPP|DLSW|SDLC|UNdef)  
[ActLimit=<limit(1-16)>] [DatMode=(Half|Full)] [ROle=(Neg|Pri|Sec)]
```

With this command, you set the data link control (DLC) type and other attributes for the port. Specify LLC2 for token ring, Ethernet, FDDI, or Boundary Access Node (BAN) links. Specify PPP for PPP links. Specify FR for Frame Relay links for Boundary Network Node (BNN), DLSW for DLSw links, or SDLC for SDLC links.

If you specify DLSW as the DLC type, or if you specify LLC2 for BAN links only, you must specify the port number as !0.

For more information about the PortDef parameter, see the SNA Service Parameters chapter in *Reference for Enterprise OS Software*.

- 3 If you set the port DLC type in step 2 to LLC2, FR, or DLSw, go to step a. If you set the port DLC type to SDLC in step 2, go to step b.
 - a Define the SSCP link station to a port using:

```
ADD !<port> -SNA SscpLinkSta <pu name> <dest media addr> [sap=<num>]  
[LinkName=name] [AutoStart=(Yes|No)] [Xid3=(Yes|No)]
```

Define a local PU that will use the link to communicate with the SSCP. If the DLC type specified with the PortDef parameter is LLC2 or DLSw, the destination address is a MAC address and SAP. If the DLC type is Frame Relay, the address will be a DLCI. If you specified !0 in the previous step for BAN links, specify !0

for the SscpLinkSta parameter. If you set the AutoStart value to No, then you must start the link station or initiate the link from the host side.

For more information about the PortDef parameter, see the SNA Service Parameters chapter in *Reference for Enterprise OS Software*.

- b** Define the SSCP link station to a port over an SDLC line using:

```
ADD !<port> -SNA SdLcLinkSta <pu name> <station addr>(Hex 1-FE)
  [LinkName=name] [AutoStart =(Yes|No)] [Xid3=(Yes|No)]
  [SendWindow=<num>] [ContactTimer=<num>] [NoRspTimer=<num>]
  [NoRsptimRetry=<num>]
```

If the port DLC type set in step 2 is PPP, you do not need to specify a media address.

- 4** For BSC Conversion, you can optionally define LUs that are used to convert BSC to SNA using:

```
ADD -SNA LUDef <luname> <nau>(1-254) <puname> [Model=(2|3|4|5|Unknown)]
```



The luname only has local significance however, it should match the LU name on the host for network diagnosis. Also, the model designation is only used when dynamic definition of dependent LUs (DDDLU) is being used.

- 5** Repeat step 3 for each SSCP or SDLC link station added.

You can add up to 16 SSCP link stations or 16 SDLC link stations to a port.

- 6** If you configured multiple SSCP-PU sessions to different hosts, define the default PU using:

```
SETDefault -SNA DefaultPU <pu name>
```

The DefaultPU parameter is required when applications are added that support the sending of unsolicited ALERTS. The PU name must match one of the PU names defined with either the SscpLinkSta or SdLcLinkSta parameters.

- 7** Enable the SNA Service by entering:

```
SETDefault -SNA CONTROL = Enable
```

After this command has been enabled, the bridge/router can communicate with the VTAM host.



If you use DLSw, you need another route at the other end. Also note that the IDBLK and IDNUM must be the same on both hosts.

Figure 273 shows a sample configuration, and Table 66 lists the commands required to configure both the bridge/router and the VTAM hosts so that the SSCP-PU sessions can take place.

Figure 273 SSCP-PU Session Configuration

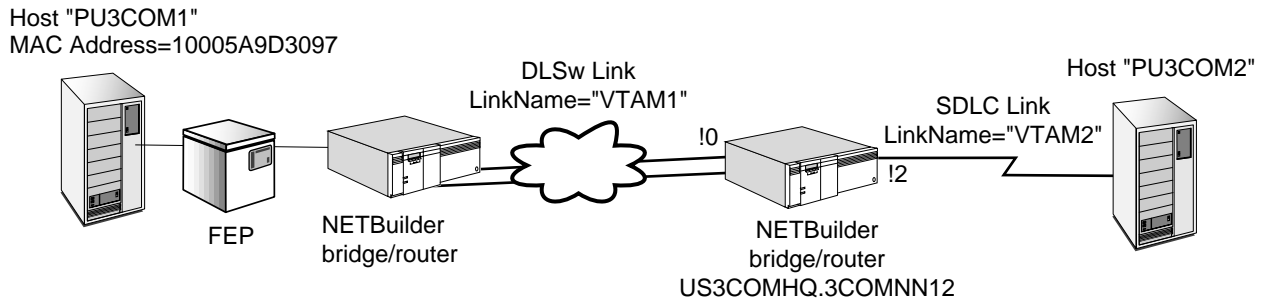


Table 66 SSCP-PU Session Configuration Commands

| NETBuilder Bridge/Router | VTAM Host |
|---|---|
| SETDefault -SNA LocalNodeName
US3COMHQ.PU3COM2 01724001 | Configuration on VTAM |
| SETDefault !0 -SNA PortDef = DLSW | PU3COM 1 PU ADDR=01 |
| | XID:
IDBLK: 017
IDNUM: 24001 |
| ADD !0 -SNA SscpLinkSta PU3COM1
10005A9D3097 LinkName=VTAM
Xid3=Yes | LU31HB12 LU LOCADDR=
LU31HB13 LU LOCADDR=3 |
| ADD -SNA LUDef LU31HB12 2 PU3COM1 | |
| ADD -SNA LUDef LU31HB13 3 PU3COM1 | |
| SETDefault -SNA CONTROL = Enable | |



For Local Node xxxx the NetID should match the NetID the host is using. The xxxx can match the PUname, but it does not have to.

Activating and Deactivating SSCP Link Stations

You can dynamically activate and deactivate a link to a specific SSCP link station using:

```
SET -SNA LinkStaCONT = <linkname> Activate|Deactivate
```

For example, to deactivate the link LINK0004, enter:

```
SET -SNA LinkStaCONT = LINK0004 Deactivate
```

To reactivate that link, enter:

```
SET -SNA LinkStaCONT = LINK0004 Activate
```

To determine the link name (if assigned by the system), enter:

```
SHOW -SNA LinkStaCONT
```

Activating and Deactivating All SSCP-PU Sessions

You can dynamically activate and deactivate the SNA Service, which affects all SSCP-PU sessions, using:

```
SET -SNA CONTROL = Enable | Disable
```

For example, to dynamically deactivate the SNA Service, which brings down all SSCP-PU sessions, enter:

```
SET -SNA CONTROL = Disable
```

To reactivate the SNA Service, enter:

```
SET -SNA CONTROL = Enable
```

After you reactivate the SNA Service, SSCP-PU sessions automatically come up only if AutoStart is set to Yes on the link stations.

Checking LU Status

To check the status of the LUs, enter:

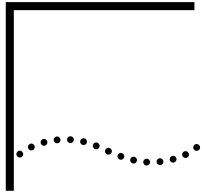
```
SHOW -SNA LUSTATUS
```

A display similar to the following appears:

```

-----LU
Status-----
PU Name    LU Name    Addr      SSCP-LU    PLU-SLU
PU3Com1    LU31HB12  2         Active     Active
PU3Com1    LU31HB13  3         Active     Inactive

```



CONFIGURING BINARY SYNCHRONOUS COMMUNICATIONS CONNECTIVITY

This chapter describes how to configure the bridge/router to provide Binary Synchronous Communications (BSC, also known as BISYNC) connectivity over DLSw networks and how to configure BSC for conversion to SNA. Using BSC pass-through, you can enable a secondary BSC control unit (CU) at a remote site to access a primary BSC device/host at a central site using a DLSw connection across the WAN. Using conversion, BSC can be used to access SNA applications



BSC is supported only on selected models of the SuperStack II NETBuilder bridge/router and the OfficeConnect NETBuilder bridge/router. Also, BSC pass-through is supported on leased lines only.

The bridge/router supports the following BSC protocols:

- BSC 3270 protocol
- BSC 3780/2780 protocol (point-to-point and pass through only)

The BSC protocols supported can be used on the following IBM platforms:

- 3X74 controllers
- 3780/2780 RJE
- 3745 front end processors

This BSC implementation only supports EBCDIC versions of BSC.

BSC does not support local switching, which means BSC traffic cannot be received on one port and then transmitted out another port as pure BSC traffic. The BSC traffic must be transmitted out the second port over a DLSw connection.

Configuring BSC Pass-Through

To configure BSC pass-through for a single CU accessing a host, you need to configure BSC for the central site and remote site bridge/routers. The following procedure describes how to configure both.

For information on the BSC Service parameters used in these procedures, see the BSC Service Parameters chapter in *Reference for Enterprise OS Software*.

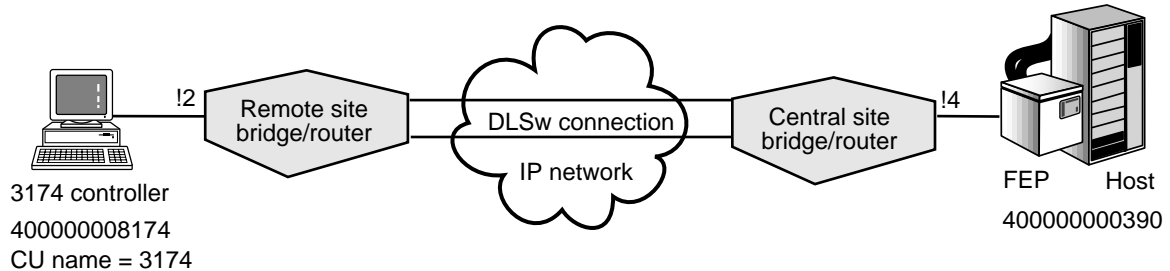
Prerequisites

Before beginning these procedures, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/routers according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the DLSw connection over the WAN between the central site bridge/router and remote site bridge/router using the procedures described in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter and the Configuring Multicast Data Link Switching for NetBIOS and SNA Networks chapter.

Figure 274 shows a configuration in which a single CU at a remote site (secondary) is accessing a host (primary) at a central site using BSC that is transmitted across the WAN through a DLSw connection.

Figure 274 BSC Single Secondary CU Configuration



Remote Site Configuration

To configure the remote site bridge/router, follow these steps:

- 1 For the port connected to the BSC device, configure the port owner to BSC using:

```
SETDefault !<port> -PORT OWNeR = BSC
```

- 2 Set internal clocking on the path using:

```
SETDefault !<path> -PATH CLock = Internal
```

Because the BSC device is a DTE, the bridge/router must act as a DCE and provide internal clocking, or alternatively, you can use modem eliminators to provide clocking. If you change the bridge/router from acting as a DTE to acting as a DCE, you must use a different cable. For more information, see the *WAN Cabling and Connectivity Guide*. You can find this guide on the 3Com Corporation World Wide Web site by entering: <http://www.3com.com/>.

- 3 Set the baud rate on the path using:

```
SETDefault !<path> -PATH BAud = <kbps> (0.110-16000)
```

Only the following baud rates are supported for BSC:

| | | |
|-----|------|------|
| 1.2 | 1.8 | 2.4 |
| 3.6 | 4.8 | 7.2 |
| 9.6 | 19.2 | 38.4 |

The baud rates set for both the local and remote site bridge/routers must match or be close so that the BSC host will receive responses in a timely manner. For more information, see "Baud Rate and Line Speed Considerations" later in this chapter.

- 4 Set the path line type for leased using:

```
SETDefault !<path> -PATH LineType = Leased
```

BSC pass-through is supported on leased lines only.

- 5 Toggle the path using:

```
SETDefault !<path> -PATH CONTrol = Enable
```

Toggling the path enables the clocking and baud rate settings to take effect.



Other PATH Service parameters such as DUplex, ENCOding, and TxDle do not need to be configured, and their values are ignored by BSC.

For information about parameters in the PATH Service, see the PATH Service Parameters chapter in *Reference for Enterprise OS Software*.

- 6 Set the port as a BSC primary device using:

```
SETDefault !<port> -BSC Role = Primary
```

The port on the remote site bridge/router must be set to Primary because the BSC devices are always secondary devices.

For example, to configure port 2 as the BSC primary, enter:

```
SETDefault !2 -BSC Role = Primary
```



CAUTION: Do not configure both the remote site and central site as primary. If both sides of the BSC link are configured to primary, neither side will initiate the DLSw circuit, and no BSC traffic will be sent.

- 7 Enable BSC on the port using:

```
SETDefault !<port> -BSC CONTrol = Enable
```

- 8 Define the BSC CU that represents the BSC controller and enable it using:

```
ADD !<port> -BSC BscCU <cu name> <cu addr> <local mac> <remote mac>
[lsap=<value>] [rsap=<value>] [ENable]
```

The name can be up to 8 characters long, must be unique on the bridge/router, and it cannot be the name "ALL." The name is not case-sensitive. The CU name is used to define the CU information on the bridge/router, and is also used to disable and enable the CU for modifying the CU definition (see "Modifying Existing BSC CU Definitions" later in this chapter).

The CU address must be between 0 and 31. The MAC addresses must be entered in noncanonical format and must be in the valid LAA range (see the Configuring LAN Address Administration chapter).

For example, to define the CU named "3174" on port 2 with a CU address of 10 and local MAC address of 400000003174 and a remote MAC address of 400000000390, and use the default SAP values, enter:

```
ADD !2 -BSC BscCU 3174 10 400000003174 400000000390 ENable
```

The specified CU at the remote site is ready for the BSC connection, which can take place after BSC is configured on the central site bridge/router.

Central Site Configuration

To configure BSC on the central site bridge/router, follow these steps:

- 1 Configure the port attached to the front-end processor (FEP) for BSC using:

```
SETDefault !<port> -PORT OWNer = BSC
```

- 2 Set external clocking on the path using:

```
SETDefault !<path> -PATH CLock = External
```

Set the path to External clocking because most FEPs provide clocking. Consult your systems programmer to verify that the FEP provides clocking on the line. If the FEP does not provide clocking, set the -PATH CLock parameter to Internal and use the appropriate cables for the bridge/router and the FEP; if you use internal clocking, you must also set the baud rate using the SETDefault !<path> -PATH BAud command.

- 3 Set the path line type for leased using:

```
SETDefault !<path> -PATH LineType = Leased
```

BSC pass-through is supported on leased lines only.

4 Toggle the path using:

```
SETDefault !<path> -PATH CONTROL = Enable
```

Toggleing the path enables the clocking and connector parameters to take effect.

5 Set the port as a BSC secondary device using:

```
SETDefault !<port> -BSC Role = Secondary
```

The port on the central site bridge/router must be set to Secondary because the FEP or host is always a primary device.

For example, to set port 4 as the BSC secondary, enter:

```
SETDefault !4 -BSC Role = Secondary
```



CAUTION: Do not configure both the central site and remote site as secondary. If both sides of the BSC link are configured to secondary, neither side will initiate the DLSw circuit, and no BSC traffic will be sent.

6 Enable BSC on the port using:

```
SETDefault !<port> -BSC CONTROL = Enable
```

7 Define the BSC CU that represents the CU at the remote site and enable it using:

```
ADD !<port> -BSC BscCU <cu name> <cu addr> <local mac> <remote mac>
[Rsap=<value>] [Rsap=<value>] [ENable]
```

The MAC addresses must be entered in noncanonical format and must be in the valid LAA range (see the Configuring LAN Address Administration chapter).

The local and remote MAC addresses should be the reverse of the local and remote MAC addresses entered in step 8 in the remote site bridge/router procedure. If you define the local and remote SAP values, you should also enter the reverse SAP values that you configured on the remote site.

For example, to define and enable a CU named "3174" on port 4 with a CU address of 10 and local MAC address of 40000000390 and a remote MAC address of 400000003174, enter:

```
ADD !4 -BSC BscCU 3174 10 40000000390 400000003174 ENable
```

The BscCU definition entered here refers to the device at the remote site, even though it is added on the host port.

The remote site BSC can access the central site host (assuming that the DLSw connection across the WAN is correctly configured).

8 Repeat step 7 for each CU you will connect to at the central site.

Baud Rate and Line Speed Considerations

Because BSC is a time-sensitive protocol, you should be careful when configuring the DLSw connection baud rate so that BSC traffic can be effectively transmitted across the network. Note the following considerations when configuring BSC:

- The baud rate configured for the central site BSC link should match the baud rate for the remote site BSC link or be close to it. If there is a wide variance between the baud rates on both bridge/routers, BSC transmission errors and time-outs can occur.
- The baud rate configured for the DLSw connection across the WAN must be higher than the baud rate configured for the BSC links. To prevent BSC session time-outs, follow these guidelines:
 - If you are running only BSC traffic across the DLSw connection, the baud rate for the DLSw connection across the WAN must be higher than the baud rate for the corresponding BSC links.

- If you are running BSC traffic with other IBM traffic types, such as SDLC and LLC2, over the same DLSw connection, you must increase the link speed of the DLSw connection and use DLSw prioritization to prioritize the BSC traffic higher than the non-BSC traffic. For more information about DLSw prioritization, see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.
- If you are running BSC traffic with non-IBM traffic types such as IPX over the same DLSw connection, you must increase the link speed of the DLSw connection and you must use either protocol reservation or data prioritization to prioritize the BSC traffic higher than the other traffic types. You must set BSC traffic to a higher priority so that the BSC traffic will maintain enough speed to ensure proper request/response between the BSC devices. For more information, see the Configuring Protocol Reservation chapter and the Prioritizing Multiprotocol Data chapter.

Modifying Existing BSC CU Definitions

After you have added and defined a BSC CU, you can modify the CU definition. You must first disable the CU using:

```
SETDefault !<CU name> -BSC CUCONTROL = Disable
```

To modify an existing BSC CU definition, use:

```
ADD !<port> -BSC BscCU <cu name> <cu addr> <local mac> <remote mac>  
[Lsap=<value>] [Rsap=<value>] [ENable]
```

You can enable the CU using the ENable option when you change the definition. If you do not want to enable the CU at that time, do not specify ENable. You can reenable the CU at another time without entering the definition using:

```
SETDefault !<CU name> -BSC CUCONTROL = Enable
```

You can delete a single defined CU or all defined CUs on a port using:

```
DElete !<port> -BSC BscCU <cu name> | ALL
```

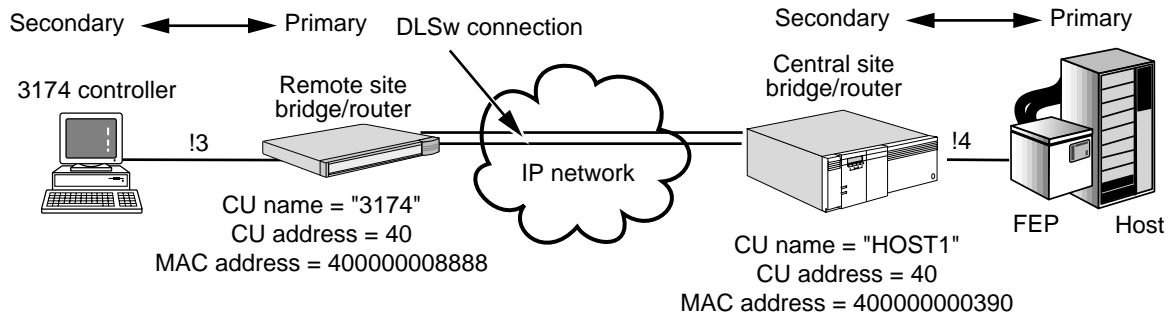
You can only delete CUs that have been disabled with the CUCONTROL parameter.

BSC Configuration Examples

This section provides two BSC configuration examples, one for configuring a single CU on a remote site, and one for configuring multiple CUs at a remote site.

Example 1: CU At Single Remote Site

Figure 275 is a BSC configuration example for a single CU at the remote site. Table 67 lists the commands necessary to configure BSC on both the remote site and central site bridge/routers.

Figure 275 BSC Configuration Example (Single CUs)**Table 67** BSC Configuration Example Commands (Single CU)

| Commands Entered at Remote Site | Commands Entered at Central Site |
|--|--|
| SETDefault !3 -PORT OWNEr = BSC | SETDefault !4 -PORT OWNEr = BSC |
| SETDefault !3 -PATH CLock = Internal | SETDefault !4 -PATH CLock = External |
| SETDefault !3 -PATH BAud = 9.6 | SETDefault !4 -PATH BAud = 9.6 |
| SETDefault !3 -PATH LineType = Leased | SETDefault !4 -PATH LineType = Leased |
| SETDefault !3 -PATH CONTRol = Enable | SETDefault !4 -PATH CONTRol = Enable |
| SETDefault !3 -BSC Role = Primary | SETDefault !4 -BSC Role = Secondary |
| SETDefault !3 -BSC CONTRol = Enable | SETDefault !4 -BSC CONTRol = Enable |
| ADD !3 -BSC BscCU 3174 40
400000008888 400000000390
ENable | ADD !4 -BSC BscCU 3174 40
400000000390 400000008888
ENable |



Certain commands shown in the table use the port ID mapped to the path ID. On most bridge/router models, the port and path numbers are mapped one-to-one. On bridge/router models with ISDN interfaces, the default path number mapped to the port number is one number different; for example path 4 is mapped to port 3.

Example 2: Multiple CUs On One Port at a Remote Site

Figure 280 is a BSC configuration example for multiple CUs on one remote site port. Table 68 lists the commands necessary to configure BSC on both the remote site and central site bridge/routers.

Figure 276 Multiple CUs on One Remote Site Port

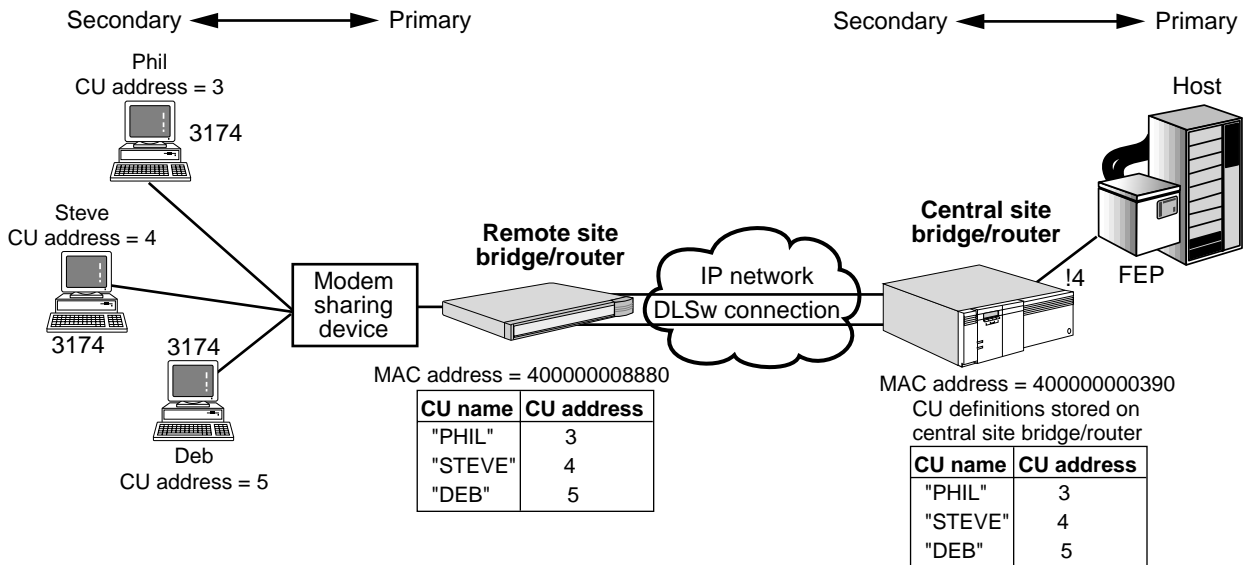


Table 68 BSC Configuration Example Commands (Multiple CUs on One Port)

| Commands Entered at Remote Site | Commands Entered at Central Site |
|--|--|
| SETDefault !3 -PORT OWNer = BSC | SETDefault !4 -PORT OWNer = BSC |
| SETDefault !3 -PATH CLock = Internal | SETDefault !4 -PATH CLock = External |
| SETDefault !3 -PATH BAud = 9.6 | SETDefault !4 -PATH BAud = 9.6 |
| SETDefault !3 -PATH LineType = Leased | SETDefault !4 -PATH LineType = Leased |
| SETDefault !3 -PATH CONTROL = Enable | SETDefault !4 -PATH CONTROL = Enable |
| SETDefault !3 -BSC Role = Primary | SETDefault !4 -BSC Role = Secondary |
| SETDefault !3 -BSC CONTROL = Enable | SETDefault !4 -BSC CONTROL = Enable |
| ADD !3 -BSC BscCU PHIL 3
400000008880 400000000390
ENable | ADD !4 -BSC BscCU PHIL 3
400000000390 400000008880
ENable |
| ADD !3 -BSC BscCU STEVE 4
400000008880 400000000390
ENable | ADD !4 -BSC BscCU STEVE 4
400000000390 400000008880
ENable |
| ADD !3 -BSC BscCU DEB 5
400000008880 400000000390
ENable | ADD !4 -BSC BscCU DEB 5
400000000390 400000008880
ENable |

Example 3: CUs at Multiple Remote Sites

Figure 277 is a BSC configuration example for a virtual multidrop environment (multiple remote sites, each with one CU). Table 69 lists the commands necessary to configure BSC on both the remote site and central site bridge/routers.

Figure 277 BSC Virtual Multidrop Configuration Example

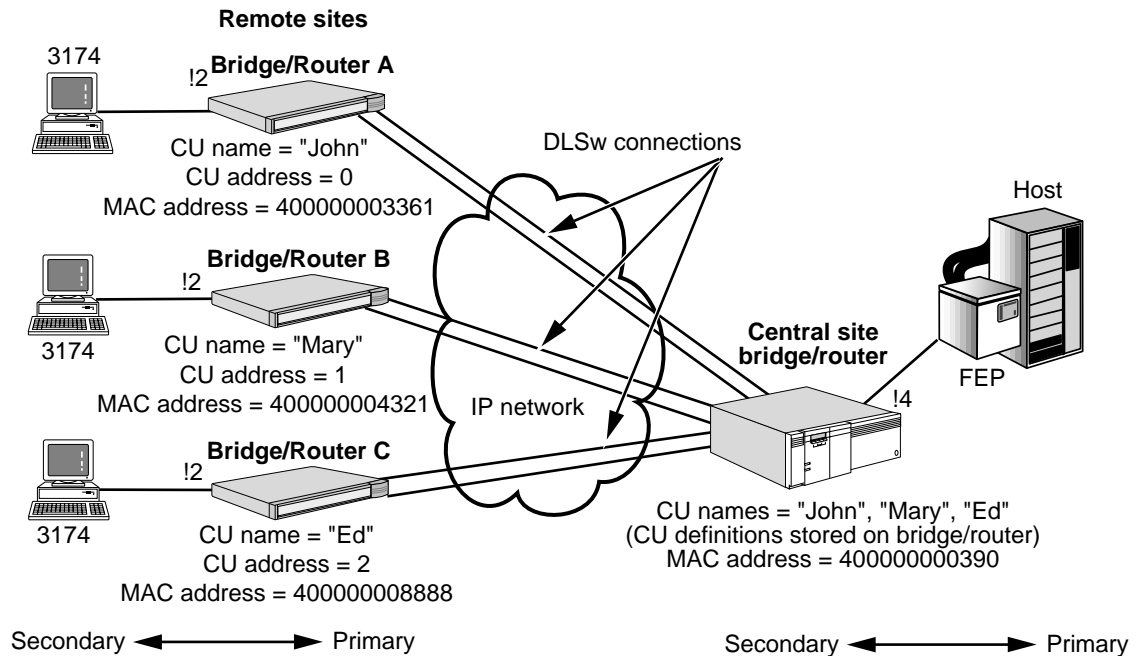


Table 69 BSC Configuration Example Commands (Multiple CUs)

| Commands Entered at Remote Sites | Commands Entered at Central Site |
|---|--|
| <u>Remote Site A:</u> | |
| SETDefault !2 -PORT OWNer = BSC | SETDefault !4 -PORT OWNer = BSC |
| SETDefault !2 -PATH CLock = Internal | SETDefault !4 -PATH CLock = External |
| SETDefault !2 -PATH BAud = 9.6 | SETDefault !4 -PATH LineType = Leased |
| SETDefault !2 -PATH LineType = Leased | SETDefault !4 -PATH CONTROl = Enable |
| SETDefault !2 -PATH CONTROl = Enable | SETDefault !4 -BSC Role = Secondary |
| SETDefault !2 -BSC Role = Primary | SETDefault !4 -BSC CONTROl = Enable |
| SETDefault !2 -BSC CONTROl = Enable | ADD !4 -BSC BscCU JOHN 0
400000000390 400000003361 ENable |
| ADD !2 -BSC BscCU JOHN 0
400000003361 400000000390
ENable | ADD !4 -BSC BscCU MARY 1
400000000390 400000004321 ENable |
| <u>Remote Site B:</u> | ADD !4 -BSC BscCU ED 2
400000000390 400000008888 ENable |
| SETDefault !2 -PORT OWNer = BSC | |
| SETDefault !2 -PATH CLock = Internal | |
| SETDefault !2 -PATH BAud = 9.6 | |
| SETDefault !2 -PATH LineType = Leased | |

Table 69 BSC Configuration Example Commands (Multiple CUs) (continued)

| Commands Entered at Remote Sites | Commands Entered at Central Site |
|---|----------------------------------|
| SETDefault !2 -PATH CONTROL = Enable | |
| SETDefault !2 -BSC Role = Primary | |
| SETDefault !2 -BSC CONTROL = Enable | |
| ADD !2 -BSC BscCU MARY 1
400000004321 400000000390
ENable | |
| <u>Remote Site C:</u> | |
| SETDefault !2 -PORT OWNER = BSC | |
| SETDefault !2 -PATH CLock = Internal | |
| SETDefault !2 -PATH BAud = 9.6 | |
| SETDefault !2 -PATH LineType = Leased | |
| SETDefault !2 -PATH CONTROL = Enable | |
| SETDefault !2 -BSC Role = Primary | |
| SETDefault !2 -BSC CONTROL = Enable | |
| ADD !2 -BSC BscCU ED 2
400000008888 400000000390
ENable | |

Configuring BSC Conversion

To configure BSC conversion for a single CU accessing a host, you need to configure BSC and SNA for conversion to work properly.

For information on the BSC Service parameters used in these procedures, see the BSC Service Parameters chapter in *Reference for Enterprise OS Software*.

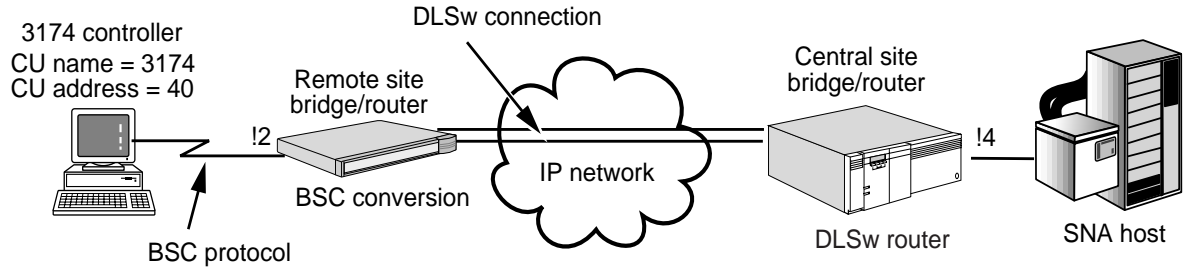
Prerequisites

Before beginning these procedures, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/routers according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- If using DLSw, configure your bridge/router according to the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.
- If using SNA, configure your bridge/router according to the Configuring NetView Service Point chapter.

Figure 278 shows a configuration in which a single CU at a remote site is accessing an SNA host at a central site.

Figure 278 BSC Conversion Single Secondary CU Configuration



BSC Conversion Configuration

To configure BSC conversion, follow these steps:

- 1 For the port connected to the BSC device, configure the port owner to BSC using:

```
SETDefault !<port> -PORT OWNeR = BSC
```

- 2 Set internal clocking on the path using:

```
SETDefault !<path> -PATH CLock = Internal
```

Because the BSC device is a DTE, the bridge/router must act as a DCE and provide internal clocking, or alternatively, you can use modem eliminators to provide clocking. If you change the bridge/router from acting as a DTE to acting as a DCE, you must use a different cable. For more information, see the *WAN Cabling and Connectivity Guide*. You can find this guide on the 3Com Corporation World Wide Web site by entering: <http://www.3com.com/>.

- 3 Set the baud rate on the path using:

```
SETDefault !<path> -PATH BAud = <kbps> (0.110-16000)
```

Only the following baud rates are supported for BSC:

| | | |
|-----|------|------|
| 1.2 | 1.8 | 2.4 |
| 3.6 | 4.8 | 7.2 |
| 9.6 | 19.2 | 38.4 |

- 4 Set the path line type for leased using:

```
SETDefault !<path> -PATH LineType = Leased
```

BSC conversion is supported on leased lines only.

- 5 Toggle the path using:

```
SETDefault !<path> -PATH CONTroL = Enable
```

Toggling the path enables the clocking and baud rate settings to take effect.



Other PATH Service parameters such as ENCoding, and TxIdle do not need to be configured, and their values are ignored by BSC.

For information about parameters in the PATH Service, see the PATH Service Parameters chapter in *Reference for Enterprise OS Software*.

- 6 Set the duplex to match the device type. If modems are used you must specify the duplex to half using:

```
SETDefault !<port> -PAth Duplex = Half
```


- 7 Set the port as a BSC Conversion Primary port using:

```
SETDefault !<port> -BSC Role = ConversionPrimary
```

For example, to configure port 2 as the conversion primary, enter:

```
SETDefault !2 -BSC Role = ConversionPrimary
```

- 8 Define the BSC CU that represents the BSC controller and enable it using:

```
ADD !<port> -BSC BscCU <cu name> <cu addr> [ENable]
```

The name can be up to eight characters long, must be unique on the bridge/router, and it cannot be the name "ALL." The name is not case-sensitive. The CU name is used to define the CU information on the bridge/router, and is also used to disable and enable the CU for modifying the CU definition (see "Modifying Existing BSC CU Definitions" earlier in this chapter).

The CU address must be between 0 and 31. EBCDIC values may also be used. See Table 72 for BSC CU to EBCDIC value mapping.

For example, to define the CU named "3174" on port 2 with a CU address of 10, enter:

```
ADD !2 -BSC BscCU 3174 10 ENable
```

The specified CU at the remote site is ready for the BSC connection, which can take place after BSC is configured on the central site bridge/router.

- 9 Define the BSC devices for the CU by entering:

```
ADD !2 - BSC BscDev 10 0 luxyz
```

```
ADD !2 -BSC BscDev 10 1 luabc
```

If you do not specify a device name, a name will be generated automatically. In the example, the device 0 (EBCDIC 40) on CU 10 (EBCDIC 4A) on port 2 will use luxyz. Device 1 (EBCDIC C1) on the same CU will use luabc.



The LUs luxyz and luabc must be defined in the SNA Service.

- 10 Multiple CUs are supported on the same BSC port. Repeat steps 9 and 10 as necessary for your configuration.
- 11 Enable BSC on the port using:

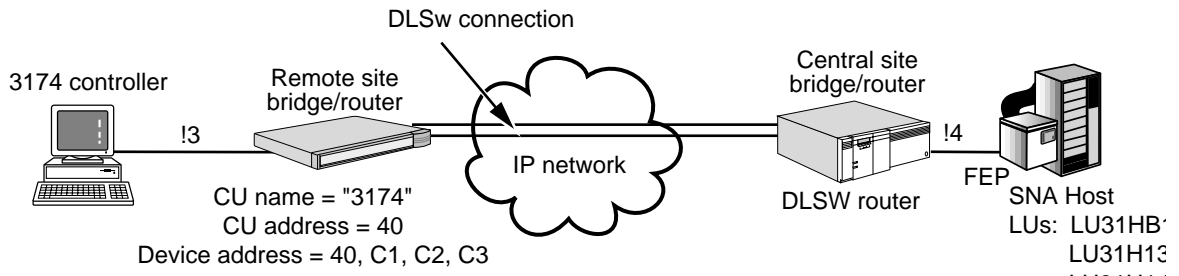
```
SETDefault !<port> -BSC CONTrol = Enable
```

BSC Conversion Examples

This section provides two BSC configuration examples, one for configuring a single CU on a remote site, and one for configuring multiple CUs at a remote site.

Example 1: Single CU With Multiple Devices At Single Remote Site

Figure 279 is a BSC conversion configuration example for a single CU at the remote site. Table 70 lists the commands necessary to configure BSC on the remote router.

Figure 279 BSC Conversion Configuration Example (Single CUs)

The remote bridge/router is always a BSC primary when performing BSC conversion.

Table 70 BSC Configuration Example Commands (Single CU)

Commands Entered at Remote Site

```

SETDefault !3 -PORT OWNer = BSC
SETDefault !3 -PATH CLock = Internal
SETDefault !3 -PATH BAud = 9.6
SETDefault !3 -PATH LineType = Leased
SETDefault !3 -PATH CONTROL = Enable
SETDefault !3 -PATH Duplex = Half
SETDefault !3 -BSC Role = ConversionPrimary
SETDefault !3 -BSC CONTROL = Enable
ADD !3 -BSC BscCU 3174 40 ENable
ADD !3 -BSC BscDev 40 40 LU31HB12
ADD !3 -BSC BscDev 40 C1 LU31H13
ADD !3 -BSC BscDev 40 C2 LU31H14
ADD !3 -BSC BscDev 40 C3 LU31H15

```



Certain commands shown in the table use the port ID mapped to the path ID. On most bridge/routers, the port and path numbers are mapped one-to-one. On bridge/routers with ISDN interfaces, the default path number mapped to the port number is one number different; for example path 4 is mapped to port 3.

Example 2: Multiple ATM CUs On One Port at a Remote Site

Figure 280 is a BSC Conversion configuration example for multiple ATM CUs on one remote site port. Table 71 lists the commands necessary to configure BSC the remote router.

Figure 280 BSC Configuration Example (Multiple CUs on One Port)

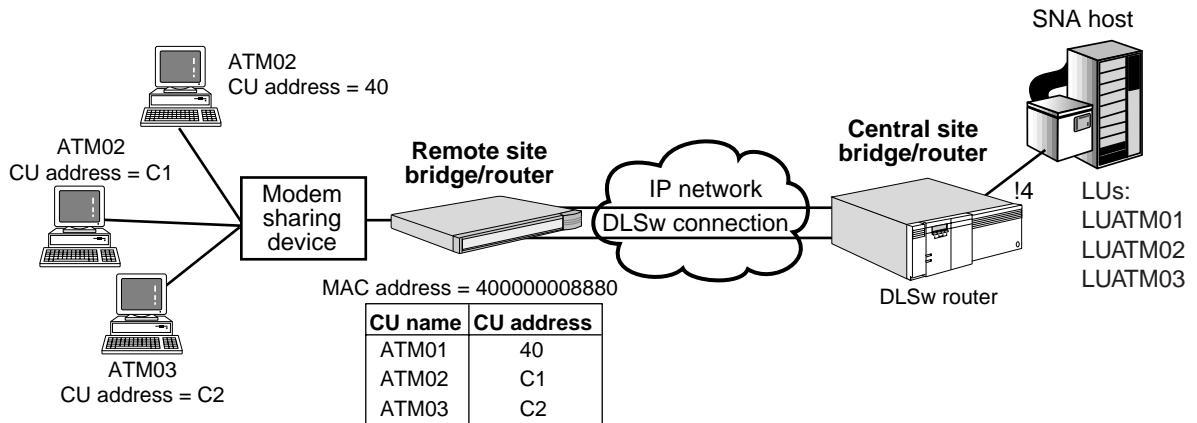


Table 71 BSC Configuration Example Commands (Multiple CUs on One Port)

Commands Entered at Remote Site

```

SETDefault !3 -PORT OWNer = BSC
SETDefault !3 -PATH CLock = Internal
SETDefault !3 -PATH BAud = 9.6
SETDefault !3 -PATH LineType = Leased
SETDefault !3 -PATH DUplex = Half
SETDefault !3 -PATH CONTROl = Enable
SETDefault !3 -BSC Role = ConversionPrimary
SETDefault !3 -BSC CONTROl = Enable
ADD !3 -BSC BscCU ATM01 40 ENable AtmMode = Yes
ADD !3 -BSC BscCU ATM01 C1 ENable AtmMode = Yes
ADD !3 -BSC BscCU ATM02 C2 ENable AtmMode = Yes
ADD !3 -BSC BscDev 40 40 LUATM01
ADD !3 -BSC BscDev C1 40 LUATM02
ADD !3 -BSC BscDev C2 40 LUATM03
    
```



For most ATMs, the Device Address should be EBCDIC 40 (or Device 0). If this is not the case, the device address used should match what the ATM expects.



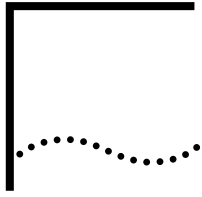
The LUs must be configured in the SNA Service for the BSC conversion to work properly.

Table 72 BSC CU/Device Address Mapping Table

| Device/CU Address | EBCDIC (hex) Value | Device/CU Address | EBCDIC (hex) Value |
|-------------------|--------------------|-------------------|--------------------|
| 0 | 40 | 16 | 50 |
| 1 | C1 | 17 | D1 |
| 2 | C2 | 18 | D2 |
| 3 | C3 | 19 | D3 |
| 4 | C4 | 20 | D4 |
| 5 | C5 | 21 | D5 |
| 6 | C6 | 22 | D6 |
| 7 | C7 | 23 | D7 |
| 8 | C8 | 24 | D8 |
| 9 | C9 | 25 | D9 |
| 10 | 4A | 26 | 5A |
| 11 | 4B | 27 | 5B |
| 12 | 4C | 28 | 5C |
| 13 | 4D | 29 | 5D |
| 14 | 4E | 30 | 5E |
| 15 | 4F | 31 | 5F |



For CU addresses, this is the polling address NOT the selection address.



CONFIGURING POLLED ASYNCH CONNECTIVITY

This chapter describes how to configure the bridge/router to provide polled asynchronous communications (referred to as *asynch* in the remainder of this chapter) so that remote asynch devices can communicate with an asynch polling host. The bridge/router offers transparent transmission of asynch “pass-through” across the WAN.



Polled asynch connectivity is supported only on selected models of the SuperStack II NETBuilder bridge/router and the OfficeConnect NETBuilder bridge/router. Polled asynch is only supported on RS232 ports.

Some parameters provided can be used to configure the tunnel framing appropriate for the asynchronous protocol and devices being used. These protocols are vendor-specific. For more information about the specific asynchronous protocol being used, check your vendor's documentation.

Configuring Asynch Tunnels on Both Central and Remote Sites

To configure polled asynch connectivity, you need to configure asynch for the local and the remote bridge/router. The following procedure describes how to configure both bridge/routers.

For information on the ATUN Service parameters used in these procedures, see the ATUN Service Parameters chapter in *Reference for Enterprise OS Software*.

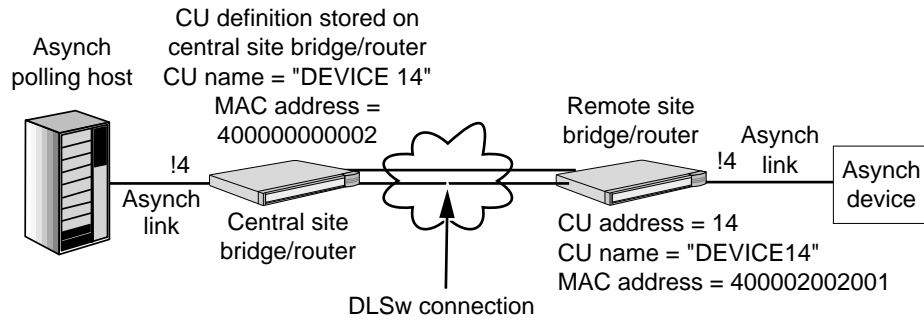
Prerequisites

Before beginning these procedures, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/routers according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the DLSw connection over the WAN between the central site bridge/router and remote site bridge/routers using the procedures described in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter and the Configuring Multicast Data Link Switching for NetBIOS and SNA Networks chapter.

Figure 281 shows a configuration in which a single asynch device CU at a remote site is communicating with an asynch polling host at a central site with a DLSw connection across the WAN. Note that you can have multiple CUs at a central site, but only one CU on a remote site port.

This example provides a procedure for a simple configuration. For more specific configuration examples, see “Asynch Tunneling Configuration Examples” later in this chapter.

Figure 281 Asynch Configuration with a Single CU

The configuration procedure is organized by the following major tasks (some parameters may be optional):

- General asynch port and path configuration:
 Set the `-Port OWNEr` parameter to `ATUN`, and configure the `-PATH BAud`, `DataBits`, `StopBits`, and `PARity` parameters, and re-enable the `-PATH CONTRol` parameter
- Asynch port configuration:
 Configure the `-ATUN FrameSize`, `FrameChars`, `IdleTimer`, `FrameGap`, `PortCONTRol`, `CUAddress`, and `AddrLOCation` parameters
- Asynch CU configuration:
 Configure the `-ATUN LocalMac`, `LocalSap`, `RemoteMac`, and `RemoteSap` parameters, and enable the `-ATUN CUCONTRol` parameter.

The procedures are performed on both the central site and remote site bridge/routers.

General Asynch Port and Path Configuration

The first major task is to configure asynch support on port and paths. To configure asynch port and path support, perform the following steps on both the central and remote site bridge/routers:

- 1 For the port connected to the asynch device, configure the port owner to `ATUN` by entering:

```
SETDefault !<port> -PORT OWNEr = ATUN
```

- 2 Set the baud rate on the path using:

```
SETDefault !<path> -PATH BAud = <kbps> (0.110-16000)
```

Only the following baud rates are supported for asynch:

| | | | | |
|-------|-------|-------|-------|-------|
| 0.110 | 0.135 | 0.150 | 0.200 | 0.300 |
| 0.600 | 1.2 | 1.8 | 2.4 | 3.6 |
| 4.8 | 7.2 | 9.6 | 19.2 | 38.4 |

When entering a baud rate less than 1.0, you must enter the leading 0 before the decimal point (for example, 0.110, not .110). For baud rates with a trailing zero (for example, 0.150), you do not have to enter the trailing 0.

For example, to set the baud rate to 0.600 on path 4, enter:

```
SETDefault !4 -PATH BAud = 0.6
```



The baud rate configured on the bridge/router must be consistent with the baud rate configured on the asynch device.

- 3 Configure the transmission characteristics of the asynch path by performing the following sub-steps:
 - a Configure the number of data bits in each character transmitted or received on the asynch path using:


```
SETDefault !<path> -PATH DataBits = 5 | 6 | 7 | 8
```
 - b Configure the number of stop bits appended to each character transmitted on the asynch path using:


```
SETDefault !<path> -PATH StopBits = 1 | 1.5 | 2
```
 - c Configure how you want the parity bits appended to each transmitted or received character using:


```
SETDefault !<path> -PATH PARity = Even | Odd | Mark | Space | None
```

Using this parameter, you specify whether the parity bit is appended to make the total parity even or odd. Or, you can specify whether the parity bit appended is always 1 (Mark), or 0 (Space). This parameter applies to both transmitted and received characters. To configure different values for transmitted and received characters, use the -PATH RxParity and -PATH TxParity features. For more information about these parameters, see the PATH Service Parameters chapter in *Reference for Enterprise OS Software*.
- 4 Toggle the path using:


```
SETDefault !<path> -PATH CONTrol = Enable
```

toggling the path enables the baud rate and the other -PATH Service parameters to take effect.

Proceed to the next section.

Asynch Port Configuration

The next steps determine how the bridge/router groups received data into frames for transmission, and how these frames are routed to asynch tunnels. The correct setting of these -ATUN Service parameters will depend on the specific asynch protocol you are using. For more information about parameters in the ATUN Service, see the ATUN Service Parameters chapter in *Reference for Enterprise OS Software*.

To configure asynch ports, follow these steps:

- 1 Configure how the incoming character stream is broken into frames by performing the following steps:
 - a Configure the maximum number of bytes to be collected before forwarding using:


```
SETDefault !<port> -ATUN FrameSize = <bytes> (1-1024)
```

This parameter may be used to reduce latency for an application with a fixed frame size by saving the wait for IdleTimer expiration.
 - b Configure the special characters that will indicate the end of a frame using:


```
SETDefault !<port> -ATUN FrameChars <char>...
```
 - c Configure the length of idle time interval that will cause accumulated data to be forwarded as a frame using:


```
SETDefault !<port> -ATUN IdleTimer = <milliseconds> (0-5000)
```

- d Configure the minimum amount of idle time to leave between frames transmitted by the bridge/router using:

```
SETDefault !<port> -ATUN FrameGap = <milliseconds> (0-1000)
```

If the FrameGap parameter is configured, the bridge/router separates the frames before sending them out the port.

- 2 To configure how the asynch port will be used, perform the following steps:

- a On the central site, configure the asynch port using:

```
SETDefault !<port> -ATUN PortCONTROL = ([Enabled | Disabled],
    [CentralSite | RemoteSite], [Address | NoAddress], [BCAddr |
    NoBCAddr], [ForcePoll | NoForcePoll], [TestEcho | NoTestEcho])
```

Specify CentralSite and Enabled to enable the asynch port. When you configure the asynch port for the central site, you can set the port to provide addressing using the Address value.

For example, to enable asynch on port 4 and set it for central site operation and to enable addressing on the port, enter:

```
SETDefault !4 -ATUN PortCONTROL = Enabled, CentralSite, Address
```



Although addressing is not required on the central site, it is recommended where possible. Note that the addressing parameters may need to be different at opposite ends of the tunnel.

If you choose to use addressing, you can choose whether to use the address specified with the -ATUN BroadCastAddr parameter by specifying the BCAddr value.

- b On the remote site, configure the asynch port using:

```
SETDefault !<port> -ATUN PortCONTROL = ([Enabled | Disabled],
    [CentralSite | RemoteSite], [Address | NoAddress], [BCAddr |
    NoBCAddr], [ForcePoll | NoForcePoll], [TestEcho | NoTestEcho])
```

Specify RemoteSite and Enabled to enable the asynch port. For example, to enable asynch port 4 and set it for remote site operation and for no addressing (recommended for remote sites), enter:

```
SETDefault !4 -ATUN PortCONTROL = Enabled, RemoteSite, NoAddress
```

The NoAddress setting specifies that each frame is sent on every asynch tunnel. A remote site port allows only a single CU (tunnel), so addressing is generally not necessary. By not configuring specific addresses on the remote site, the remote site configuration is simpler than the central site configuration.

For more information about the PortCONTROL parameter, see the ATUN Service Parameters chapter in *Reference for Enterprise OS Software*.

- 3 If you specified the Address value in the previous step (normally for central sites only), configure addressing by performing the following sub-steps:

- a Configure the address location using:

```
SETDefault !<port> -ATUN AddrLOCation = <offset> (0-1024)
```

This parameter specifies which data byte of received frames should be considered an address byte. The value is specified as an offset from the first byte of the frame.

For example, to set the offset on port 4 to 1, enter:

```
SETDefault !4 -ATUN AddrLOCation = 1
```


- b** If you specified the BCAddr value in the PortCONTROL parameter, configure the broadcast address using:

```
SETDefault !<port> -ATUN BroadCastAddr = <value> (0-255)
```

The address you enter is a special value for the address byte and indicates an “all stations” destination. All frames whose address byte (as specified with AddrLOCATION) matches the value of the broadcast address are forwarded to all active asynch tunnels on the port.

For example, to configure the broadcast address for port 4 to 255, enter:

```
SETDefault !4 -ATUN BroadCastAddr = 255
```

Proceed to the next section.

Asynch CU Configuration

The following steps define CUs on the bridge/router to provide the definition of the CUs, which determine the tunnel connection(s) to and from remote sites. To configure asynch CUs, follow these steps:

- 1 Define the name of each CU and assign each one to the port using:

```
ADD !<port> -ATUN PortCU <CU name>...
```

Each CU name can be up to 8 characters long and must be unique on the bridge/router. Use the CU name(s) to configure the remaining parameters in the central site procedure.

The difference between defining CUs on a central site and a remote site is:

- On a central site port, you can define multiple CUs, each representing a tunnel to a remote site.
- On a remote site port, you can define a single CU, which provides a tunnel endpoint to the central site.

For example, to define the CU name “DEVICE14” on port 4, enter:

```
SETDefault !4 -ATUN PortCU DEVICE14
```

You can define multiple CUs on a central site using one command. For example, if there are two other remote site CUs named DEVICE15 and DEVICE16, you can define all three on port 4 by entering:

```
ADD !4 -ATUN PortCU DEVICE14 DEVICE15 DEVICE16
```

- 2 Define the CU address using:

```
SETDefault !<CU name> -ATUN CUADDRESS = <value>(0-255) [--<value>(0-255)]
```

For example, to assign a CU address of 1 to a CU named HOST1, enter:

```
SETDefault !HOST1 -ATUN CUADDRESS = 1
```

If you have multiple physical devices, multidropped at a remote site, only a single tunnel is configured to a remote port using a single CU definition. In this case, you can use an address range to route frames for all the devices into the same tunnel. For example, if you have four CUs with CU addresses of 15, 16, 17, and 18, you can specify a CU address range of 15-18 and assign it to a CU name of DEVICE15 by entering:

```
SETDefault !DEVICE15 -ATUN CUADDRESS = 15-18
```

When addressing is used on the port (as set with the PortCONTROL parameter), an address byte is extracted from each frame (set with the AddrLOCATION parameter). The frame is then directed to the CU whose address range includes this value.

The CUADDRESS values for all enabled CUs on a port cannot overlap; an addressed frame is mapped to a single CU only.

- 3 Configure the peer MAC addresses for the tunnel endpoints by performing the following steps on both the central site and remote site bridge/routers:
 - a On the central site bridge/router, configure the MAC address of the CU on the central site that will be used as the source address for initiating a DLSw circuit or LLC2 session using:

```
SETDefault !<CU name> -ATUN LocalMac = <address>
```

By default, the address is configured in noncanonical format and must be in the valid LAA range (see the Configuring LAN Address Administration chapter). You can configure the address in canonical format by entering the prefix "mac" or "cmac" before the address. The MAC address entered for the LocalMac parameter must be unique; the same MAC address cannot be used as the value for the LocalMac parameter on the same asynch network.

For example, to configure the local MAC address for the CU named DEVICE14, enter:

```
SETDefault !DEVICE14 -ATUN LocalMac = 400000000002
```

The MAC address you enter as the LocalMac is the same address you will enter in step d as the RemoteMac address on the remote site bridge/router.

- b On the central site bridge/router, configure the MAC address of the CU on the remote site that will be used as the peer address for the DLSw circuit or LLC2 session using:

```
SETDefault !<CU name> -ATUN RemoteMac = <address>
```

By default, the address is configured in noncanonical format. The same restrictions for LocalMac described in the previous step also apply to the RemoteMac parameter.

For example, to configure the MAC address of the remote site CU, enter:

```
SETDefault !DEVICE14 -ATUN RemoteMac = 400002002001
```

- c On the remote site bridge/router, configure the MAC address of the CU on the remote site that will be used as the source address for initiating a DLSw circuit or LLC2 session using:

```
SETDefault !<CU name> -ATUN LocalMac = <address>
```

Use the same MAC address that you configured in step b. For example, to configure the local MAC address as 400002002001 for the CU named DEVICE14, enter:

```
SETDefault !DEVICE14 -ATUN LocalMac = 400002002001
```

- d On the remote site bridge/router, configure the MAC address of the CU on the central site that will be used as the peer address for the DLSw circuit or LLC2 session using:

```
SETDefault !<CU name> -ATUN RemoteMac = <address>
```

Use the same MAC address you configured in step a. For example, to configure the remote MAC address as 400000000002 for the CU named DEVICE14, enter:

```
SETDefault !DEVICE14 -ATUN RemoteMac = 400000000002
```

- 4 Optionally, configure the peer SAP values for both tunnel peers by performing the following substeps on both the central site and remote site bridge/routers:
 - a On the central site bridge/router, configure the local SAP as the source SAP for initiating a DLSw circuit for tunneling asynch data using:


```
SETDefault !<CU name> -ATUN LocalSap = <sap> (hex 04-ec[by 4])
```

 For example, to configure the local SAP as 04 for DEVICE14, enter:


```
SETDefault !DEVICE14 -ATUN LocalSap = 04
```
 - b On the central site bridge/router, configure the remote SAP as the destination SAP for initiating a DLSw circuit for tunneling asynch data using:


```
SETDefault !<CU name> -ATUN RemoteSap = <sap> (hex 04-ec[by 4])
```

 For example, to configure the remote SAP as 04 for DEVICE14, enter:


```
SETDefault !DEVICE14 -ATUN RemoteSap = 04
```
 - c Repeat steps a and b on the remote site bridge/router, reversing the SAP values entered at the central site. For example, if the SAP values are different, enter the LocalSap value entered on the central site as the RemoteSap value on the remote site, and vice-versa.
- 5 Activate the asynch connection to the CU using:


```
SETDefault !<CU name> -ATUN CUControl = (Enabled | Disabled)
```

 After you enter this parameter, the asynch connection will be ready to accept or initiate sessions.
 For example, to activate an asynch connection to the CU named DEVICE14, enter:


```
SETDefault !DEVICE14 -ATUN CUControl = Enabled
```

 Repeat this step on both the central site and remote site bridge/routers.
- 6 Repeat steps 2 through 5 for each CU name defined in step 1.

Asynch Tunneling Configuration Examples

This section provides two asynch tunneling configuration examples. The first example shows a central site communicating with three remote sites, each with a single asynch device. The second example shows a central site communicating with two remote sites, each with different and more complex asynch configurations.



The specific settings of the -PATH Service parameters depend on the devices being used. Similarly the -ATUN framing and addressing parameters must be appropriate for the devices and the protocols in use. These examples describe the characteristics of a hypothetical polling protocol.

Example 1: Single Asynch Devices at the Remote Sites

Figure 282 shows a configuration with a central site communicating with three remote sites, each with a single asynch device. Table 73 lists the commands to configure the asynch tunneling at the central site and for the three remote sites.

For this example, the asynch devices are operating at 9600 baud, using 8 databits and even parity, and needs one stop bit. The host sends fixed-size (4-byte) poll frames with an address in the third byte of every frame (no broadcast), and the devices send variable-sized frames (80-byte maximum) in response to a poll. Both the asynch host and the asynch devices send characters back-to-back within a

frame, but delay at least 30 milliseconds between sending frames; they also recognize a received frame by seeing a delay of at least 10 milliseconds.

Figure 282 Asynch Configuration Example (Single CU at Remote Sites)

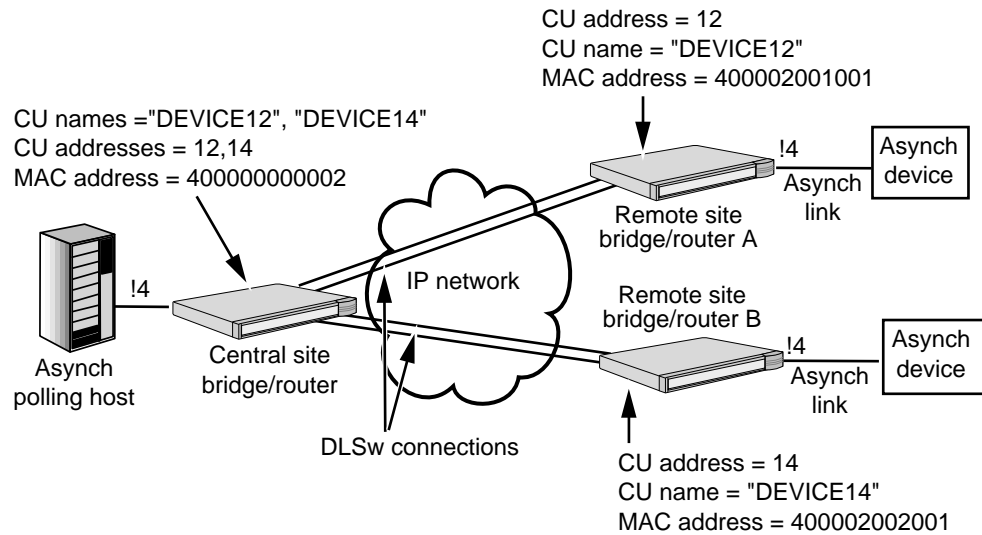


Table 73 Asynch Configuration Example Commands (Single CU at Each Remote Site)

| Commands Entered at Central Site: | Commands Entered at Remote Sites: |
|--|---|
| | <u>Remote Site A:</u> |
| SETDefault !4 -PORT OWNer = ATUN | SETDefault !4 -ATUN PORT OWNer = ATUN |
| SETDefault !4 -PATH BAud = 9.6 | SETDefault !4 -ATUN PATH BAud = 9.6 |
| SETDefault !4 -PATH DataBits = 8 | SETDefault !4 -PATH DataBits = 8 |
| SETDefault !4 -PATH PARity = Even | SETDefault !4 -PATH PARity = Even |
| SETDefault !4 -PATH StopBits = 1 | SETDefault !4 -PATH StopBits = 1 |
| SETDefault !4 -PATH CONTrol = Enable | SETDefault !4 -ATUN PATH CONTrol = Enable |
| SETDefault !4 -ATUN FrameSize = 4 | SETDefault !4 -ATUN FrameSize = 80 |
| SETDefault !4 -ATUN IdleTimer = 10 | SETDefault !4 -ATUN IdleTimer = 10 |
| SETDefault !4 -ATUN FrameGap = 10 | SETDefault !4 -ATUN FrameGap = 10 |
| SETDefault !4 -ATUN PortCONTrol = (Enabled,
CentralSite, Address, NoBCAddr, NoForcePoll,
NoTestEcho) | SETDefault !4 -ATUN PortCONTrol = (Enabled,
RemoteSite, NoAddress, NoBCAddr, NoForcePoll,
NoTestEcho) |
| SETDefault !4 -ATUN AddrLOCation = 2 | ADD !4 -ATUN PortCU = DEVICE12 |
| ADD !4 -ATUN PortCU = DEVICE12 DEVICE14 | SETDefault !DEVICE12 -ATUN CUADDRess = 12 |
| SETDefault !DEVICE12 -ATUN CUADDRess = 12 | SETDefault !DEVICE12 -ATUN LocalMac =
400002001001 |
| SETDefault !DEVICE12 -ATUN LocalMac =
400000000002 | SETDefault !DEVICE12 -ATUN RemoteMac =
400000000002 |
| SETDefault !DEVICE12 -ATUN RemoteMac =
400002001001 | SETDefault !DEVICE12 -ATUN CUCONTrol = Enabled |
| SETDefault !DEVICE12 -ATUN CUCONTrol = Enabled | <u>Remote Site B:</u> |
| SETDefault !DEVICE14 -ATUN CUADDRess = 14 | SETDefault !4 -ATUN PORT OWNer = ATUN |

Table 73 Asynch Configuration Example Commands (Single CU at Each Remote Site) (continued)

| Commands Entered at Central Site: | Commands Entered at Remote Sites: |
|---|---|
| (continued) | |
| SETDefault !DEVICE14 -ATUN LocalMac = 400000000002 | SETDefault !4 -ATUN PATH BAud = 9.6 |
| SETDefault !DEVICE14 -ATUN RemoteMac = 400002002001 | SETDefault !4 -PATH DataBits = 8 |
| SETDefault !DEVICE14 -ATUN CUCONTROL = Enabled | SETDefault !4 -PATH PARity = Even |
| | SETDefault !4 -PATH StopBits = 1 |
| | SETDefault !4 -ATUN PATH CONTROL = Enable |
| | SETDefault !4 -ATUN FrameSize = 80 |
| | SETDefault !4 -ATUN IdleTimer = 10 |
| | SETDefault !4 -ATUN FrameGap = 10 |
| | SETDefault !4 -ATUN PortCONTROL = (Enabled, RemoteSite, NoAddress, NoForcePoll, NoBCAddr) |
| | ADD !4 -ATUN PortCU = DEVICE14 |
| | SETDefault !DEVICE14 -ATUN CUADDRESS = 14 |
| | SETDefault !DEVICE14 -ATUN LocalMac = 400002002001 |
| | SETDefault !DEVICE14 -ATUN RemoteMac = 400000000002 |
| | SETDefault !DEVICE14 -ATUN CUCONTROL = Enabled |

Example 2: Multiple Asynch Devices at Remote Sites

Figure 283 shows a configuration with a central site communicating with two remote sites, each with a different configuration. The bridge/router at remote site A has multiple attached asynch devices, each one over a separate port. The bridge/router at remote site B is connected to a modem sharing device, which is connected to multiple asynch devices; since only one tunnel is allowed to that port (a remote site port only allows a single CU definition), an address range is used in the CU definition at the central site. The CU range allows all three devices to map to a single tunnel.

Table 74 lists the commands to configure the asynch tunneling at both the central site and for the three remote sites.

For this example, the asynch host is operating at 9600 baud, using 7 databits. The asynch host is transmitting even parity and receiving odd parity and expecting two stop bits. The asynch devices are similar but transmit odd and receive even parity. The host sends variable length frames, each with an address in the first byte, and the special address 255 is meant to go to all devices (broadcast). The devices send variable-sized frames in response that always terminate with an ASCII Carriage Return (decimal 13).

Both the asynch host and the asynch devices send characters back-to-back within a frame, but delay at least 30 milliseconds between frames. When receiving frames, the asynch host and asynch devices do not depend on the inter-frame gap to recognize a frame.

Figure 283 Asynch Configuration Example (Multiple CUs at Remote Sites)

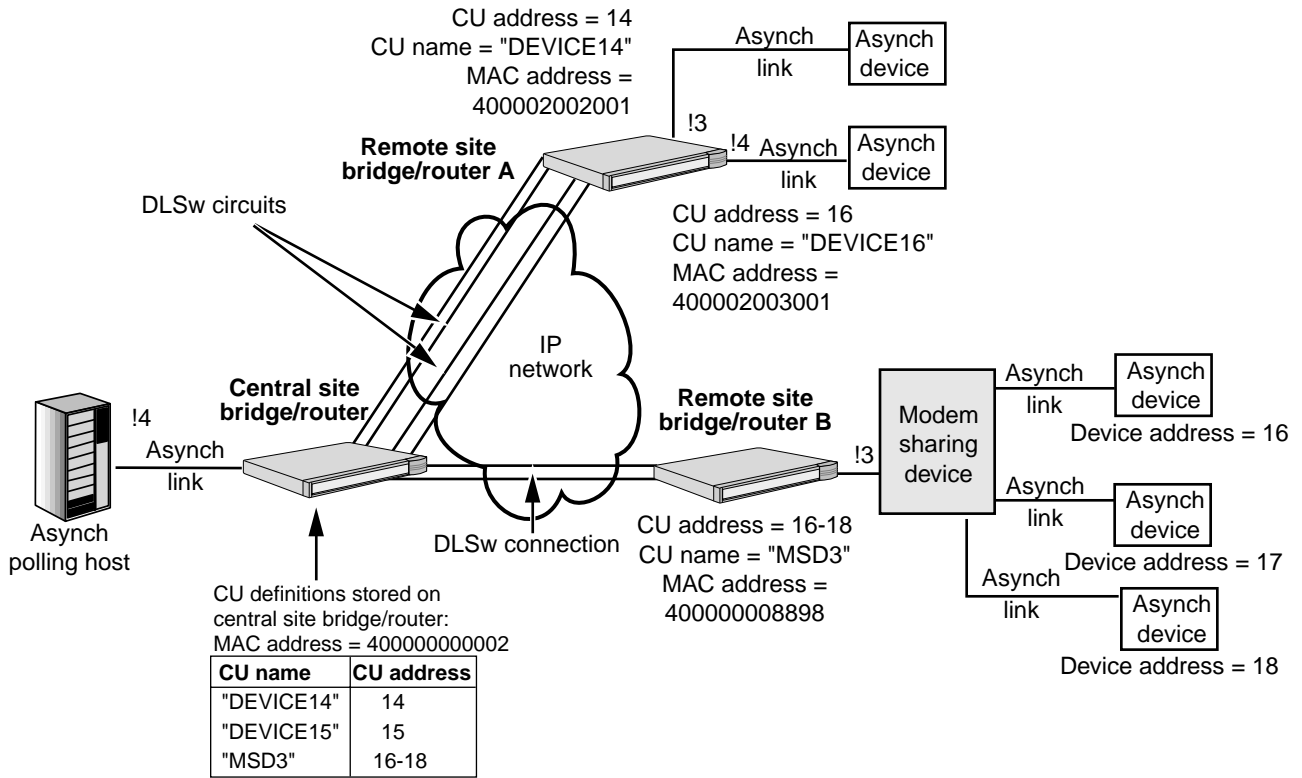
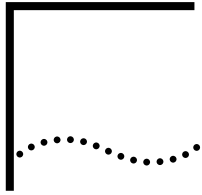


Table 74 Asynch Configuration Example Commands (Multiple CUs at Remote Sites)

| Commands Entered at Central Site: | Commands Entered at Remote Sites: |
|--|--|
| | <u>Remote Site A:</u> |
| SETDefault !4 -PORT OWNer = ATUN | SETDefault !3 -PORT OWNer = ATUN |
| SETDefault !4 -PATH BAud = 9.6 | SETDefault !3 -PATH BAud = 9.6 |
| SETDefault !4 -PATH DataBits = 7 | SETDefault !3 -PATH DataBits = 7 |
| SETDefault !4 -PATH RxParity = Even | SETDefault !3 -PATH RxParity = Odd |
| SETDefault !4 -PATH TxParity = Odd | SETDefault !3 -PATH TxParity = Even |
| SETDefault !4 -PATH StopBits = 2 | SETDefault !3 -PATH StopBits = 2 |
| SETDefault !4 -PATH CONTrol = Enable | SETDefault !3 -PATH CONTrol = Enable |
| SETDefault !4 -ATUN IdleTimer = 20 | ADD !3 -ATUN FrameChars 13 |
| SETDefault !4 -ATUN FrameGap = 0 | SETDefault !3 -ATUN IdleTimer = 20 |
| SETDefault !4 -ATUN PortCONTrol = (Enabled,
CentralSite, Address, BCAddr, NoForcePoll,
NoTestEcho) | SETDefault !3 -ATUN FrameGap = 0 |
| SETDefault !4 -ATUN AddrLOCation = 0 | SETDefault !3 -ATUN PortCONTrol = (Enabled, RemoteSite,
NoAddress, NoBCAddr, NoForcePoll, NoTestEcho) |
| SETDefault !4 -ATUN BroadCastAddr = 255 | SETDefault !4 -PORT OWNer = ATUN |
| ADD !4 -ATUN PortCU = DEVICE14 DEVICE15
MSD3 | SETDefault !4 -PATH BAud = 9.6 |
| SETDefault !DEVICE14 -ATUN CUADDRess = 14 | SETDefault !4 -PATH DataBits = 7 |
| SETDefault !DEVICE14 -ATUN LocalMac =
400000000002 | SETDefault !4 -PATH RxParity = Odd |
| SETDefault !DEVICE14 -ATUN RemoteMac =
400002002001 | SETDefault !4 -PATH TxParity = Even |
| (continued) | |
| SETDefault !DEVICE14 -ATUN CUCONTrol =
Enabled | SETDefault !4 -PATH StopBits = 2 |
| SETDefault !DEVICE15 -ATUN CUADDRess = 16 | SETDefault !4 -PATH CONTrol = Enable |
| SETDefault !DEVICE15 -ATUN LocalMac =
400000000002 | ADD !4 -ATUN FrameChars 13 |
| SETDefault !DEVICE15 -ATUN RemoteMac =
400002003001 | SETDefault !4 -ATUN IdleTimer = 20 |
| SETDefault !DEVICE15 -ATUN CUCONTrol =
Enabled | SETDefault !4 -ATUN FrameGap = 0 |
| SETDefault !MSD3 -ATUN CUADDRess = 16-18 | SETDefault !4 -ATUN PortCONTrol = (Enabled, RemoteSite,
NoAddress, NoBCAddr, NoForcePoll, NoTestEcho) |
| SETDefault !MSD3 -ATUN LocalMac =
400000000002 | ADD !3 -ATUN PortCU = DEVICE14 |
| SETDefault !MSD3 -ATUN RemoteMac =
400000008898 | SETDefault !DEVICE14 -ATUN LocalMac = 400002002001 |
| SETDefault !MSD3 -ATUN CUCONTrol = Enabled | SETDefault !DEVICE14 -ATUN LocalSap = 04 |
| | SETDefault !DEVICE14 -ATUN RemoteMac =
400000000002 |

Table 74 Asynch Configuration Example Commands (Multiple CUs at Remote Sites) (continued)

| Commands Entered at Central Site: | Commands Entered at Remote Sites: |
|-----------------------------------|--|
| | SETDefault !DEVICE14 -ATUN CUCONTRol = Enabled
ADD !4 -ATUN PortCU = DEVICE16
SETDefault !DEVICE15 -ATUN LocalMac = 400002003001
SETDefault !DEVICE15 -ATUN LocalSap = 04
SETDefault !DEVICE15 -ATUN RemoteMac = 400000000002
SETDefault !DEVICE15 -ATUN CUCONTRol = Enabled |
| | <u>Remote Site B:</u>
SETDefault !4 -ATUN PORT OWNEr = ATUN
SETDefault !4 -ATUN PATH BAud = 9.6
SETDefault !4 -ATUN PATH DataBits = 7
SETDefault !4 -PATH RxParity = Odd
SETDefault !4 -PATH TxParity = Even
SETDefault !4 -PATH StopBits = 2
SETDefault !4 -PATH CONTRol = Enable
SETDefault !4 -ATUN FrameChars 13
SETDefault !4 -ATUN IdleTimer = 20
SETDefault !4 -ATUN FrameGap = 0
SETDefault !4 -ATUN PortCONTRol = (Enabled, RemoteSite, NoAddress, NoBCAddr, NoForcePoll, NoTestEcho)
ADD !3 -ATUN PortCU = MSD3
SETDefault !MSD3 -ATUN LocalMac = 400000008898
SETDefault !MSD3 -ATUN RemoteMac = 400000000002
SETDefault !MSD3 -ATUN CUCONTRol = Enabled |



CONFIGURING BOUNDARY ROUTING SYSTEM ARCHITECTURE

This chapter describes how to implement Boundary Routing system architecture, how Boundary Routing works, and where it can be used.

The information in this chapter applies to Boundary Routing system architecture in both non-IBM and IBM environments except where specifically called out. A *non-IBM environment* is an environment where Systems Network Architecture (SNA) or NetBIOS are not used, for example, an Internet Protocol (IP) or Internetwork Packet Exchange (IPX) environment. An *IBM environment* is an SNA or NetBIOS environment.



For conceptual information, see "How Boundary Routing System Architecture Works" later in this chapter. For information on using Integrated Services Digital Network (ISDN) systems in a Boundary Routing topology, see the Configuring Wide Area Networking Using ISDN chapter.

After you configure Boundary Routing, you can set up auto startup phase 2 on the central node. For information on auto startup, see the Configuring Autostartup chapter.

Configuring Basic Boundary Routing

This section describes how to configure a wide area port on the central node for Boundary Routing over the Point-to-Point Protocol (PPP), Frame Relay, or X.25. For information on which NETBuilder platforms can be used as a central node, see "Where Can Boundary Routing Be Used?" later in this chapter.

Prerequisites

Before performing one of the following procedures, make sure that you have configured the wide area port and path. For information on configuring wide area ports and paths, see the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

The next section contains the procedure for configuring PPP. For Frame Relay configuration procedures, see "Configuring for Frame Relay." For X.25 configuration procedures, see "Configuring for X.25."

Configuring for PPP

To configure the Boundary Routing port for PPP, follow these steps:

- 1 Verify that the owner of the wide area port is PPP by entering:

```
SHow -PORT CONFIguration
```

If the port owner is not PPP, reconfigure the owner using:

```
SETDefault !<port> -PORT OWNer = PPP
```

- 2 Enable Boundary Routing using:

```
SETDefault !<port> -BCN CONTRol = Enabled
```

- 3 If you are configuring a NETBuilder II bridge/router, verify that the configured and actual media types of the remote LAN network match use:

```
SHow !<port> -BCN RemoteLanType
```

The resulting display shows the configured LAN type and the actual detected LAN type as shown in the example below:

```
Port !V2 RemoteLanType =Ethernet (Actual = Token Ring)
```

The Central Node determines the Leaf Nodes actual RemoteLanType by sending an SNMP request over the WAN link to the Leaf Node.

If the Central Node has not completed determining the actual remote LAN type and you issue another RemoteLan Type command you may receive the following message:

```
RemoteLanType command is Active Waiting for SNMP Response, Try Later
```

Several conditions such as an incorrect physical connection, a down port, a down path or a down switch may exist that would cause a Leaf Node to respond incorrectly or not at all. If the Central Node does not receive the Leaf Node's SNMP response before the Central Nodes's SNMP/UDP time out, the following result is displayed:

```
Port !v2 RemoteLanType = Ethernet (Actual = Unable to Determine)
```

If the configured and actual media types do not match, reconfigure the media type. For example, set the media type to token ring using:

```
SETDefault !<port> -BCN RemoteLanType = TokenRing
```

- 4 If you are configuring a NETBuilder II bridge/router and the port you are configuring is connected to a token ring peripheral node, set the MAC address format for Address Resolution Protocol (ARP) to noncanonical using:

```
SETDefault !<port> -PORT ProtMacAddrFmt = NonCanonARP
```

- 5 If you are planning to use the IPX Protocol in your Boundary Routing topology, determine if you want to use smart filtering.



Do not use smart filtering unless you have a stable WAN link.

You can use smart filtering to eliminate NetWare Routing Information Protocol (NRIP) and Service Advertising Protocol (SAP) broadcasts, and protocol island traffic. For conceptual information, see "Reduced WAN Usage Costs" later in this chapter.

By default, smart filtering is disabled on all ports. You can enable smart filtering using:

```
SETDefault !<port> -BCN CONTrol = SmartFiltering
```

For smart filtering to operate on a SuperStack II NETBuilder boundary router, you must also run Boundary Routing software version 7.0 or later.

After enabling the port later in this procedure, a short delay occurs before smart filtering begins filtering packets.

If you plan to configure NRIP/SAP updates on the central node to be incremental (-NRIP CONTrol = NoPeriodic and -SAP CONTrol = NoPeriodic), then smart filtering is not needed. You cannot obtain additional bandwidth savings. The use of nonperiodic NRIP and SAP updates assumes that servers on leaf networks can operate in a nonperiodic environment.

- 6 If you are configuring Boundary Routing in an IBM environment, follow these steps:
- a Enable the Boundary Routing of IBM traffic using:


```
SETDefault !<port> -BCN CONTROL = IbmTraffic
```

 For information on how Boundary Routing in an IBM environment works, see “How Boundary Routing System Architecture Works” later in this chapter.
 - b By default, the central node is configured to perform Boundary Routing of SNA traffic only. If you want the central node to perform Boundary Routing of NetBIOS traffic also, enter:


```
SETDefault -DLSw CONTROL = EnableNetBios
```
 - c Enable LLC2 on all LAN ports of the central node using:


```
SETDefault !<port> -LLC2 CONTROL = Enable
```

 If your topology includes clients on leaf networks that must exchange data, use this same syntax on the wide area ports of your central node that interface these leaf networks. For conceptual information on this topic, see “Peer Data Exchange” later in this chapter.
 - d Configure the logical link control, type 2 (LLC2) data link interface on all LAN ports of the central node, if necessary.
 For more information, see the Configuring the LLC2 Data Link Interface chapter. In most cases, the default settings of the -LLC2 parameters are sufficient and minimal (if any) configuration will be necessary.
 - e Verify that the appropriate type of bridging is enabled on all LAN ports of the central node.



The token ring port must be configured for transparent bridging when configuring a boundary router. This is the default setting, and must not be disabled.

On LAN ports that use Ethernet as the media type, verify that transparent bridging is enabled using:

```
SHow !<port> -BRidge TransparentBRidge
```

If the value of this parameter is not set to TransparentBRidge, use:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

On LAN ports that use token ring as the media type, verify that source route bridging is enabled and that a unique ring number has been assigned by entering:

```
SHow -SR CONFIguration
```

If source route bridging has not been enabled, use:

```
SETDefault !<port> -SR SrcRouBRidge = SrcRouBRidge
```

If a ring number has not been assigned, use:

```
SETDefault !<port> -SR RingNumber = [<number> (1-4095) | 0x<number> (1-FFF)]
```

- f If desired, enable the end system source routing of LLC2 packets, and optionally, of IP packets using:

```
SETDefault !<port> -SR RouteDiscovery = (LLC2, [IP])
```

When enabling end system source routing, you must specify the route discovery of LLC2 end system packets so that LLC2 sessions are locally terminated at both central and peripheral nodes. You can optionally specify the route discovery of IP end system packets so that you can determine the connectivity of network devices using the PING command.

- g** Reset the default virtual ring number (92) for your tunnel, if desired, using:

```
SETDefault -LLC2 TUNnelVRing = <Number>(1 - 4095)
```

- 7** Enable the port.

By default, all ports are enabled; however, you must re-enable the Boundary Routing port for the -PORT parameters you configured in earlier steps to take effect. To re-enable a port, use:

```
SETDefault !<port> -PORT CONTrol = Enabled
```

- 8** Determine if you want to use data compression over the Boundary Routing path.

If you want to use data compression, additional configuration steps are required. For information, see the Configuring Data Compression chapter. The procedure for configuring data compression on a link connecting a Boundary Routing peripheral node to a central node is exactly the same as on a link connecting two access routers.

- 9** If you want to administer IP addresses for the peripheral nodes (for Telnet and Simple Network Management Protocol (SNMP) management) from the central node, you must decide whether you want your peripheral nodes to acquire their IP addresses from a Reverse Address Resolution Protocol (RARP) or BOOTP server.

With the central node configured as a RARP server and the peripheral node configured as a RARP client (by default), the peripheral node can obtain its IP address by sending a RARP request to the central node. If you decide to use the RARP server option, follow steps a, b, and c.

By default, the peripheral node is configured as a client to a BOOTP server. The BOOTP server must exist on a network attached to the central node. The BOOTP server can be a 3Com product or a product supplied by another vendor. By configuring UDP Broadcast Helper on the central node, the central node propagates BOOTP requests from the peripheral node to the BOOTP server and obtains the peripheral node's IP address. If you decide to use the BOOTP server option, follow steps d and e.

- a** If you decide to acquire the peripheral node's IP address from a RARP server, enable the RARP server by entering:

```
SETDefault -ARP RarpCONTrol = RarpServer
```

- b** To acquire the peripheral node's IP address from a RARP server, assign an IP address to the peripheral node.

Two methods exist to assign an IP address to the peripheral node: port-to-IP address mapping and the IP Address Translation Table.

If you do not know the media access control (MAC) address of the peripheral node, use the port-to-IP address mapping. Configure the -IP RemoteAddress parameter to map an IP address to the virtual port over which the central node receives the RARP request from the peripheral node. For example, if the central node has virtual port V3 enabled for Boundary Routing, you can map an IP address to it by entering:

```
SETDefault !V3 -IP RemoteAddress = 129.213.1.1
```

To assign an IP address through the IP Address Translation Table, you need to obtain the MAC address of the peripheral node, which can be found on a label on the back of the SuperStack II NETBuilder platform.

Use the `-IP ADDRESS` parameter to add the IP address and MAC address of the peripheral node to the IP Address Translation Table. For example, to add the IP address of 129.213.1.1 and the MAC address of %080002A00890 to the peripheral node, enter:

```
ADD -IP ADDRESS 129.213.1.1 %080002A00890
```

- c To acquire the IP address of the peripheral node from a RARP server, check the setting of the `-IP ICMPReply` parameter by entering:

```
SHOW -IP CONFIGURATION
```

The `-IP ICMPReply` parameter should be set to `Mask`. The peripheral node also requests the subnet mask from the central node using an ICMP Address Mask request. The central node responds with the subnet mask configured for the interface on which the request is received. If you do not set the `ICMPReply` parameter to `Mask`, the central node will not send the subnet mask to the peripheral node, causing the IP address to be incomplete.

If the setting of this parameter is not `Mask`, enter:

```
SETDefault -IP ICMPReply = Mask
```

- d If you decide to acquire the IP address of the peripheral node from a BOOTP server, configure User Datagram Protocol (UDP) Broadcast Helper or check that UDP Broadcast Helper is configured.

For information on how to configure UDP Broadcast Helper, see the *Configuring UDP Broadcast Helper* chapter. To check UDP Broadcast Helper is configured, enter:

```
SHOW -UDPHELP CONFIGURATION
```

In the display that appears, make sure that the `-UDPHELP CONTROL` parameter is set to `Enable` and that either `BPSERVER` or UDP port number 67 appears on the Active Ports list. If one or both steps of the configuration have not been completed, you must follow the appropriate steps to make sure that UDP Broadcast Helper is completely configured.

- e To acquire the IP address of the peripheral node from a BOOTP server, assign an IP address to the peripheral node.

Two methods exist to assign an IP address to the peripheral node: port-to-IP address mapping and MAC address-to-IP address mapping. The type of BOOTP server you are using will determine which method you can use. If your BOOTP server is a 3Com product, you can use either method. If your BOOTP server is a product supplied by another vendor, you must use the MAC address-to-IP address mapping.

You must complete this step on your BOOTP server. For instructions on assigning an IP address using either of these methods, see the documentation that accompanies your BOOTP server.



If you are acquiring the IP address of the peripheral node using a BOOTP server provided by a non-3Com vendor, you must add the MAC address and IP address of the peripheral node to the database of the BOOTP server. See the documentation that accompanies your BOOTP server to determine how to do this.

- 10** After you configure the wide area port on the central node for the Boundary Routing feature, configure the central node for bridging and/or routing for each protocol used in the Boundary Routing topology.

For more information, see the bridging and routing chapters.

- 11** Determine whether the peripheral nodes require any configuration changes.

The configuration procedure can be performed at the peripheral node through an attached console or remotely through the central node using the Telnet Protocol. For additional information, see the documentation that accompanies the peripheral nodes.

To verify the configuration, see “Verifying the Configuration” later in this chapter.

You can also configure a back-up link over PPP to provide disaster recovery or bandwidth-on-demand. For conceptual information, see “Dial-up Backup Line for Disaster Recovery or Bandwidth-on-Demand” later in this chapter. If you need to provide a redundant link or route for mission-critical applications, see “Configuring Network Resiliency” later in this chapter.

After you configure the Boundary Routing software, you can set up auto startup phase 2 on the central node. For information on auto startup, see the Configuring Autostartup chapter.

Configuring for Frame Relay

Before beginning the following procedure, make sure that you have completed the steps in “Prerequisites” earlier in this chapter.

For more information, see the Configuring Wide Area Networking Using Frame Relay chapter. That chapter includes instructions on how to configure Frame Relay congestion control.

To configure the wide area and virtual ports on the central node for Boundary Routing over Frame Relay, follow these steps. For information on virtual ports, see the Configuring Advanced Ports and Paths chapter.

- 1** Verify that the owner of the wide area port is Frame Relay by entering:

```
SHoW -PORT CONFIguration
```

If the port owner is not Frame Relay, reconfigure the owner using:

```
SETDefault !<port> -PORT OWNEr = FrameRelay
```

- 2** If your Frame Relay switch supports a Local Management Interface (LMI) Protocol, verify that LMI is enabled on the wide area port using:

```
SHoW [!<port>] -FR CONFIguration
```

The software supports multiple types of LMI. See the description of the -FR CONTROL parameter in the FR Service Parameters chapter in *Reference for Enterprise OS Software* for information on the types of LMI supported.

Determine if the type of LMI specified for the -FR CONTROL parameter and the type supported by your switch match. If they do not match, you must reconfigure the LMI type using the -FR CONTROL parameter. For more information on this parameter, see the FR Service Parameters chapter in *Reference for Enterprise OS Software*.

If the switch does not support any LMI Protocol, configure the -FR CONTROL parameter using:

```
SETDefault !<port> -FR CONTroL = NoLmi
```

Specification ANSI T1.617 describes the LMI Protocol. An appendix in this specification includes Annex-D, which relates to the construction of LMI packets. NTT-LMI is the LMI Protocol supported by NTT Frame Relay switches.

- 3 Create a virtual port for each remote network that is attached to the Frame Relay cloud using:

```
ADD !<port> -PORT VirtualPort {<path> {<FR_DLCI>}}
```

For example, if you have a remote network on path 4 that uses Frame Relay DLCI 35, add virtual port V1 by entering:

```
ADD !V1 -PORT VirtualPort 4@35
```

- 4 Enable the Boundary Routing software for each virtual port associated with the path using:

```
SETDefault !<port> -BCN CONTROL = Enabled
```

For example, to enable Boundary Routing on virtual port V1, enter:

```
SETDefault !V1 -BCN CONTROL = Enabled
```

Make sure you enable the -BCN CONTROL parameter on a virtual port, not on a parent port. For information on parent ports, see the Configuring Advanced Ports and Paths chapter.

- 5 If you are configuring a NETBuilder II bridge/router, verify that the configured and actual media types of the remote LAN network match using:

```
SHow !<port> -BCN RemoteLanType
```

The resulting display shows the configured LAN type and the actual detected LAN type as shown here:

```
Port !V2 RemoteLanType =ETHERnet (Actual = Token Ring)
```

The Central Node determines the Leaf Nodes actual RemoteLanType by sending an SNMP request over the WAN link to the Leaf Node.

If the Central Node has not completed determining the actual remote LAN type and you issue another RemoteLan Type command you may receive the following message:

```
RemoteLanType command is Active Waiting for SNMP Response, Try Later
```

Several conditions such as an incorrect physical connection, a down port, a down path or a down switch may exist that would cause a Leaf Node to respond incorrectly or not at all. If the Central Node does not receive the Leaf Node's SNMP response before the Central Nodes's SNMP/UDP time out, the following result is displayed:

```
Port !v2 RemoteLanType = ETHERnet (Actual = Unable to Determine)
```

If the configured and actual media types do not match, reconfigure the media type. For example, set the media type to token ring using:

```
SETDefault !<port> -BCN RemoteLanType = TokenRing
```

- 6 If you are configuring a NETBuilder II bridge/router and the port you are configuring is connected to a token ring peripheral node, set the MAC address format for ARP to noncanonical using:

```
SETDefault !<port> -PORT ProtMacAddrFmt = NonCanonARP
```

- 7 If you are planning to use the IPX Protocol in your Boundary Routing topology, determine if you want to use smart filtering.



Do not use smart filtering unless you have a stable WAN link that uses the LMI Protocol.

You can use smart filtering to eliminate NRIP and SAP rebroadcasts, and protocol island traffic. For conceptual information, see “Reduced WAN Usage Costs” later in this chapter.

By default, smart filtering is disabled on all ports. You can enable smart filtering by using:

```
SETDefault !<port> -BCN CONTrol = SmartFiltering
```

Make sure you enable or disable smart filtering on the appropriate virtual ports to suit your needs.

After you enable each virtual port later in this procedure, a short delay occurs before smart filtering begins filtering packets.

If you plan to configure NRIP/SAP updates on the central node to be incremental (-NRIP CONTrol = NoPERiodic and -SAP CONTrol = NoPERiodic), then smart filtering is not needed. You cannot obtain additional bandwidth savings. The use of nonperiodic NRIP and SAP updates assumes that servers on leaf networks operate in a nonperiodic environment.

For smart filtering to operate on a SuperStack II NETBuilder boundary router, you must use Boundary Routing software version 7.0 or later.

- 8 If you are configuring Boundary Routing in an IBM environment, follow these steps:

- a Enable the Boundary Routing of IBM traffic using:

```
SETDefault !<port> -BCN CONTrol = IbmTraffic
```

For information on how Boundary Routing in an IBM environment works, see “How Boundary Routing System Architecture Works” later in this chapter.

- b By default, the central node is configured to perform Boundary Routing of SNA traffic only. If you also want the central node to perform Boundary Routing of NetBIOS traffic, enter:

```
SETDefault -DLSw CONTrol = EnableNetBios
```

- c Enable LLC2 on all LAN ports of the central node using:

```
SETDefault !<port> -LLC2 CONTrol = Enable
```

If your topology includes clients on leaf networks that must exchange data, use the same syntax specified above on the virtual ports that interface these leaf networks. For conceptual information on this topic, see “Peer Data Exchange” later in this chapter.

- d Configure the LLC2 data link interface on all central node LAN ports, if necessary.

For more information, see the Configuring the LLC2 Data Link Interface chapter. In most cases, the default settings of the -LLC2 parameters are sufficient and minimal (if any) configuration will be necessary.

- e Verify that the appropriate type of bridging is enabled on all LAN ports of the central node.

On LAN ports that use Ethernet as the media type, verify that transparent bridging is enabled using:

```
SHow !<port> -BRidge TransparentBRidge
```

If the value of this parameter is not set to TransparentBRidge, use:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

On LAN ports that use token ring as the media type, verify that source route bridging is enabled and that a unique ring number has been assigned by entering:

SHow -SR CONFiguration

If source route bridging has not been enabled, use:

```
SETDefault !<port> -SR SrcRouBRidge = SrcRouBRidge
```

If a ring number has not been assigned, use:

```
SETDefault !<port> -SR RingNumber = [<number>(1-4095) |  
0x<number>(1-FFF)]
```

- f** If desired, enable the end system source routing of LLC2 packets, and optionally, of IP packets using:

```
SETDefault !<port> -SR RouteDiscovery = (LLC2, [IP])
```

When enabling end system source routing, you must specify the route discovery of LLC2 end system packets so that LLC2 sessions are locally terminated at both central and peripheral nodes. You can optionally specify the route discovery of IP end system packets so that you can determine the connectivity of network devices using the PING command.

- g** Reset the default virtual ring number (92) for your tunnel, if desired, using:

```
SETDefault -LLC2 TUNnelVRing = <Number>(1 - 4095)
```

- 9** Enable each virtual port.

By default, all virtual ports are enabled; however, you must re-enable each virtual port for the -PORT parameters you configured in earlier steps to take effect. To re-enable a virtual port, use:

```
SETDefault !<port> -PORT CONTRol = Enabled
```

- 10** Determine if you want to use data compression over the Boundary Routing path.

If you want to use data compression, additional configuration steps are required. For information, see "Boundary Routing Features" later in this chapter. The procedure for configuring data compression on a link connecting a Boundary Routing peripheral node to a central node is exactly the same as on a link connecting two access routers.

- 11** If you want to administer IP addresses for the peripheral nodes (for Telnet and SNMP management) from the central node, you must decide whether you want your peripheral nodes to acquire their IP addresses from a RARP or BOOTP server.

With the central node configured as a RARP server and the peripheral node configured as a RARP client (by default), the peripheral node can obtain its IP address by sending a RARP request to the central node. If you decide to use the RARP server option, follow steps a, b, and c.

By default, the peripheral node is configured as a client to a BOOTP server. The BOOTP server must exist on a network attached to the central node. The BOOTP server can be a 3Com product or a product supplied by another vendor. By

configuring UDP Broadcast Helper on the central node, the central node propagates BOOTP requests from the peripheral node to the BOOTP server and obtains the IP address of the peripheral node. If you decide to use the BOOTP server option, follow steps d and e.

- a If you decide to acquire the IP address of the peripheral node from a RARP server, enable the RARP server by entering:

```
SETDefault -ARP RarpCONTRol = RarpServer
```

- b To acquire the peripheral node's IP address from a RARP server, assign an IP address to the peripheral node.

Two methods exist to assign an IP address to the peripheral node: port-to-IP address mapping and the IP Address Translation Table.

If you do not know the MAC address of the peripheral node, use the port-to-IP address mapping. Configure the -IP RemoteAddress parameter to map an IP address to the virtual port over which the central node receives the RARP request from the peripheral node. For example, if the central node has virtual port V3 enabled for Boundary Routing, you can map an IP address to it by entering:

```
SETDefault !V3 -IP RemoteAddress = 129.213.1.1
```

To assign an IP address through the IP Address Translation Table, you need to obtain the MAC address of the peripheral node, which can be found on a label on the back of the SuperStack II bridge/router.

Use the -IP ADDRESS parameter to add the IP address and MAC address of the peripheral node to the IP Address Translation Table. For example, to add the IP address of 129.213.1.1 and the MAC address of %080002A00890 to the peripheral node, enter:

```
ADD -IP ADDRESS 129.213.1.1 %080002A00890
```

- c To acquire the IP address of the peripheral node from a RARP server, check the setting of the -IP ICMPReply parameter by entering:

```
SHow -IP CONFIguration
```

The -IP ICMPReply parameter should be set to Mask. The peripheral node also requests the subnet mask from the central node using an ICMP Address Mask request. The central node responds with the subnet mask configured for the interface on which the request is received. If you do not set the ICMPReply parameter to Mask, the central node will not send the subnet mask to the peripheral node, causing the IP address to be incomplete.

If the setting of this parameter is not Mask, enter:

```
SETDefault -IP ICMPReply = Mask
```

- d If you decide to acquire the IP address of the peripheral node from a BOOTP server, configure UDP Broadcast Helper or make certain that UDP Broadcast Helper is configured.

For more information, see the Configuring UDP Broadcast Helper chapter. To make certain UDP Broadcast Helper is configured, enter:

```
SHow -UDPHELP CONFIguration
```

In the display that appears, make sure that the -UDPHELP CONTRol parameter is set to Enable and that either BPSERVER or UDP port number 67 appears on the Active Ports list. If one or both steps of the configuration have not been

completed, you must follow the appropriate steps to make sure that UDP Broadcast Helper is completely configured.

- e To acquire the IP address of the peripheral node from a BOOTP server, assign an IP address to the peripheral node.

Two methods exist to assign an IP address to the peripheral node: port-to-IP address mapping and MAC address-to-IP address mapping. The type of BOOTP server you are using will determine which method you can use. If your BOOTP server is a 3Com product, you can use either method. If your BOOTP server is a product supplied by another vendor, you must use the MAC address-to-IP address mapping.

You must complete this step on your BOOTP server. For instructions on assigning an IP address using either of these methods, see the documentation that accompanies your BOOTP server.



If you are acquiring the IP address of the peripheral node using a BOOTP server provided by a non-3Com vendor, you must add the MAC address and IP address of the peripheral node to the database of the BOOTP server. See the documentation that accompanies your BOOTP server to determine how to do this.

- 12 After configuring the wide area and virtual ports on the central node for the Boundary Routing feature, configure the virtual ports of the central node for bridging and/or routing for each protocol used in the Boundary Routing topology.

For more information on bridging and routing over Frame Relay, see the Configuring Wide Area Networking Using Frame Relay chapter.

- 13 Determine whether the peripheral nodes require any configuration changes.

The configuration procedure can be performed at the peripheral node through an attached console or remotely through the central node using the Telnet Protocol. For additional information, see the documentation that accompanies the peripheral nodes.

To verify the configuration, see "Verifying the Configuration" later in this chapter.

If you need to provide a redundant link or route for mission-critical applications, see "Configuring Network Resiliency" later in this chapter.

After you configure Boundary Routing, you can also set up auto startup phase 2 on the central node. For information on auto startup, see the Configuring Autostartup chapter.

Configuring for X.25 Before beginning the following steps, make sure that you have completed the steps in “Prerequisites” earlier in this chapter.

If you are configuring or are already performing Boundary Routing over X.25 and you re-enable the X.25 virtual port on your central node by entering `SETDefault !Vn -PORT CONTROL = Enabled`, you must also re-enable the X.25 path of the peripheral node. To re-enable the path, enter:

```
SETDefault !Vn -PATH CONTROL = Enabled
```

If you do not enter this command, the X.25 path of the peripheral node will remain up, but the peripheral node will not know that the X.25 virtual port of the central node has gone down. The peripheral node will continue to transmit packets to the central node, but the central node will not respond.

To configure wide area and virtual ports on the central node for Boundary Routing over X.25, follow these steps. For information on virtual ports, see the Configuring Advanced Ports and Paths chapter.

- 1 Set the owner of the wide area port to X.25 using:

```
SETDefault !<port> -PORT OWNER = X25
```

- 2 Configure each wide area port for communication with an X.25 PDN by assigning a DTE address using:

```
SETDefault !<port> -X25 X25Address = <0-9999999999999999>(1-15 digits)
```

For example, to assign a DTE address of 31102859060 to port 3, enter:

```
SETDefault !3 -X25 X25Address = 31102859060
```

- 3 Create a virtual port for each remote network that is attached to the X.25 cloud using:

```
ADD !<port> -PORT VirtualPort {<path> {<X.25 DTE>}}
```

For example, if you have a remote network on path 4 that uses X.25 DTE 31107551234, add virtual port V1 by entering:

```
ADD !V1 -PORT VirtualPort 4#31107551234
```

- 4 Enable the Boundary Routing feature on each virtual port associated with the path using:

```
SETDefault !<port> -BCN CONTROL = Enabled
```

For example, to enable Boundary Routing on virtual port V1, enter:

```
SETDefault !V1 -BCN CONTROL = Enabled
```

Make sure you enable the `-BCN CONTROL` parameter on a virtual port, not on a parent port. For information on parent ports, see the Configuring Advanced Ports and Paths chapter.

- 5 If you are configuring a NETBuilder II bridge/router, verify that the configured and actual media types of the remote LAN network match using:

```
SHow !<port> -BCN RemoteLanType
```

The resulting display shows the configured LAN type and the actual detected LAN type as shown here:

```
Port !V2 RemoteLanType =ETHERnet (Actual = Token Ring)
```

The Central Node determines the Leaf Nodes actual RemoteLanType by sending an SNMP request over the WAN link to the Leaf Node.

If the Central Node has not completed determining the actual remote LAN type and you issue another RemoteLan Type command you may receive the following message:

```
RemoteLanType command is Active Waiting for SNMP Response, Try Later
```

Several conditions such as an incorrect physical connection, a down port, a down path or a down switch may exist that would cause a Leaf Node to respond incorrectly or not at all. If the Central Node does not receive the Leaf Node's SNMP response before the Central Nodes's SNMP/UDP time out, the following result is displayed:

```
Port !v2 RemoteLanType = ETHernet (Actual = Unable to Determine)
```

If the configured and actual media types do not match, reconfigure the media type. For example, set the media type to token ring using:

```
SETDefault !<port> -BCN RemoteLanType = TokenRing
```

- 6 If you are configuring a NETBuilder II bridge/router and the port you are configuring is connected to a token ring peripheral node, set the MAC address format for ARP to noncanonical using:

```
SETDefault !<port> -PORT ProtMacAddrFmt = NonCanonARP
```

- 7 If you are planning to use the IPX Protocol in your Boundary Routing topology, determine if you want to use smart filtering.

You can use smart filtering to eliminate IPX NRIP and SAP rebroadcasts, and protocol island traffic. For conceptual information, see "Reduced WAN Usage Costs" later in this chapter.

By default, smart filtering is disabled on all ports. You can enable smart filtering using:

```
SETDefault !<port> -BCN CONTrol = SmartFiltering
```

Make sure you enable or disable smart filtering on the appropriate virtual ports to suit your needs.

After you enable each virtual port later in this procedure, a short delay occurs before smart filtering begins filtering packets.

If you plan to configure NRIP/SAP updates on the central node to be incremental (-NRIP CONTrol = NoPEriodic and -SAP CONTrol = NoPEriodic), then smart filtering is not needed. You can obtain no further bandwidth savings. The use of nonperiodic NRIP and SAP updates assumes that servers on leaf networks can operate in a nonperiodic environment.

For smart filtering to operate on a SuperStack II NETBuilder boundary router, you must use Boundary Routing software version 7.0 or later.

- 8 If you are configuring the Boundary Routing feature in an IBM environment, follow these steps:
 - a Enable the Boundary Routing of IBM traffic using:

```
SETDefault !<port> -BCN CONTrol = IbmTraffic
```

For information on how Boundary Routing in an IBM environment works, see "How Boundary Routing System Architecture Works" later in this chapter.

- b** By default, the central node is configured to perform Boundary Routing of SNA traffic only. If you also want the central node to perform Boundary Routing of NetBIOS traffic, enter:

```
SETDefault -DLsw CONTROL = EnableNetBios
```

- c** Enable LLC2 on all LAN ports of the central node using:

```
SETDefault !<port> -LLC2 CONTROL = Enable
```

If your topology includes clients on leaf networks that must exchange data, use the same syntax specified above on the virtual ports that interface these leaf networks. For conceptual information on this topic, see “Peer Data Exchange” later in this chapter.

- d** Configure the LLC2 data link interface on all central node LAN ports, if necessary.

For more information, see the Configuring the LLC2 Data Link Interface chapter. In most cases, the default settings of the -LLC2 parameters should be sufficient and little, if any, configuration should be necessary.

- e** Verify that the appropriate type of bridging is enabled on all LAN ports of the central node.

On LAN ports that use Ethernet as the media type, verify that transparent bridging is enabled using:

```
SHow !<port> -BRidge TransparentBRidge
```

If the value of this parameter is not set to TransparentBRidge, use:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

On LAN ports that use token ring as the media type, verify that source route bridging is enabled and that a unique ring number has been assigned by entering:

```
SHow -SR CONFIguration
```

If source route bridging has not been enabled, use:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```

If a ring number has not been assigned, use:

```
SETDefault !<port> -SR RingNumber = [<number>(1-4095) |  
0x<number>(1-FFF)]
```

- f** If desired, enable the end system source routing of LLC2 packets, and optionally, of IP packets using:

```
SETDefault !<port> -SR RouteDiscovery = (LLC2, [IP])
```

When enabling end system source routing, you must specify the route discovery of LLC2 end system packets so that LLC2 sessions are locally terminated at both central and peripheral nodes. For more information on local termination, see “Increased Reliability” later in this chapter. You can optionally specify the route discovery of IP end system packets so that you can determine the connectivity of network devices using the PING command.

- g** Reset the default virtual ring number (92) for your tunnel, if desired, using:

```
SETDefault -LLC2 TUNnelVRing = <Number>(1 - 4095)
```

- 9 Verify that the protocol identifier to be included in an outgoing X.25 call request is set appropriately using:

```
SHow !<port> -BCN X25ProtID
```

If the setting is inappropriate, specify a new protocol identifier using:

```
SETDefault !<port> -BCN X25ProtID = <protocol id> (octet)
```

The valid range includes 1 through 0xFF.

- 10 If you want to assign a higher priority to boundary-routed packets than to other types of traffic, prioritize traffic on the Boundary Routing port.

To assign a priority to boundary-routed packets, configure X.25 user profiles using the -PROFILE X25ProfileType parameter.

The default values of the X.25 parameters adhere to the default values of the X.25 standard. However, depending on the requirements of the X.25 switch your central node is connected to, it may be necessary to adjust values of parameters such as X25PacketSize, X25ThruputClass, and X25WindowSize. For more information on the X25ProfileType parameter and information on adjusting X.25 parameters to suit your installation, see the X25 Service Parameters chapter in *Reference for Enterprise OS Software*.

- 11 Enable each virtual port.

By default, all virtual ports are enabled; however, you must re-enable each virtual port for the -PORT parameters you configured in earlier steps to take effect. To re-enable a virtual port, use:

```
SETDefault !<port> -PORT CONTROL = Enabled
```

- 12 Determine if you want to use data compression over the Boundary Routing path.

If you want to use data compression, additional configuration steps are required. For information, see the Configuring Data Compression chapter. The procedure for configuring data compression on a link connecting a Boundary Routing peripheral node to a central node is exactly the same as on a link connecting two access routers.

- 13 If you want to administer IP addresses for the peripheral nodes (for Telnet and SNMP management) from the central node, use port-to-IP address mapping or the RARP IP Address Translation Table.

- a Enable the RARP server so that the central node can respond to RARP queries from the peripheral node by entering:

```
SETDefault -ARP RarpCONTROL = RarpServer
```

With the central node configured as the RARP server and the peripheral node configured as the RARP client (by default), the peripheral node can obtain its IP address by sending a RARP request to the central node.

- b Assign an IP address to the peripheral node.

If you do not know the MAC address of the peripheral node, you can use the port-to-IP address mapping. Configure the -IP RemoteAddress parameter to map an IP address to the port over which the central node receives the RARP request from the peripheral node. For example, if the virtual port V3 is enabled for the Boundary Routing feature, enter:

```
SETDefault !V3 -IP RemoteAddress = 129.213.1.1
```

To assign an IP address through the RARP IP Address Translation Table, you need the MAC address of the peripheral node. The MAC address of the peripheral node can be found on a label on the back of the SuperStack II bridge/router.

Use the `-IP ADDRESS` parameter to add the IP address of the peripheral node and MAC address to the IP Address Translation Table. For example, to add the peripheral node IP address of 129.213.1.1 and the MAC address of %080002A00890, enter:

```
ADD -IP ADDRESS 129.213.1.1 %080002A00890
```

- c** Check the setting of the `ICMPReply` parameter by entering:

```
SHow -IP CONFIguration
```

The `-IP ICMPReply` parameter should be set to `Mask`. The peripheral node also requests the subnet mask from the central node using an ICMP Address Mask request. The central node responds with the subnet mask configured for the interface on which the request is received. If you do not set the `ICMPReply` parameter to `Mask`, the central node will not send the subnet mask to the peripheral node, causing the IP address to be incomplete.

If the setting of this parameter is not `Mask`, enter:

```
SETDefault -IP ICMPReply = Mask
```

- 14** After you configure the wide area and virtual ports of the central node for the Boundary Routing feature, configure the virtual ports central node for bridging and/or routing for each protocol used in the Boundary Routing topology.

For more information on bridging and routing over X.25, see the *Configuring Wide Area Networking Using X.25* chapter.

- 15** Determine whether the peripheral nodes require any configuration changes.

The configuration procedure can be performed at the peripheral node through an attached console or remotely through the central node using the Telnet Protocol. For additional information, see the document that accompanies the peripheral nodes.

To verify the configuration, see “Verifying the Configuration” next.

If you need to provide a redundant route for mission-critical applications, see “Configuring Network Resiliency” later in this chapter.

Verifying the Configuration

To verify the initial configuration of your Boundary Routing ports or troubleshoot problems related to Boundary Routing over PPP, Frame Relay, or X.25, follow these steps:

- 1** Check the state of the ports by entering:

```
SHow -PORT CONFIguration
```

In the Current Port Parameters display, verify the following items:

- Under the Owner column, the owner of the wide area port is set correctly.
- Under the Ctrl column, the port or virtual port is enabled.
- Under the State column, the state is Up.

- 2 If the port state is not Up, check the state of the paths by entering:

SHoW -PATH CONFIguration

In the Current Path Parameters display, verify the following items:

- Under the Ctrl column, the wide area path is enabled.
- Under the State column, the state is Up.
- Under the Conn column, the connector type is appropriately set.
- Under the Clock column, the clock source is correct.

- 3 Verify that Boundary Routing is enabled on each port you configured for Boundary Routing by entering:

SHoW -BCN CONTrol

- 4 If you have configured the Boundary Routing of IBM traffic, determine the status of the Boundary Routing port using:

SHoW !<port> -BCN IbmStatus

In the IBM Status display, verify the following:

- Under the Port State column, the state is UP.
- Under the Status column, the status is ACTIVE.
- Under the State column, the state is RUNNING.

The status will be INACTIVE and the state will be STARTING for a short time while the port is activating.

The status will be INACTIVE and the state will be DISABLED if Boundary Routing has been improperly configured on the port. See “Configuring Basic Boundary Routing” earlier in this chapter to determine which step was improperly completed and redo it.

The state will be INACTIVE and the state will be REMOTE - UNKNOWN if the peripheral node is running a version of software that is incompatible with the software running on the central node or a problem exists with the WAN. Check the version of software that is running on the peripheral node. To determine the version of software that should be running, see the release notes. If software incompatibility is not the problem, check the cabling of the peripheral node and, if necessary, go on to the following step to further check the WAN.

- 5 If you have configured the Boundary Routing feature over PPP, verify the PPP configuration and status by entering:

SHoW -PPP STATUS

The Link Control Protocol (LCP) state and the Network Control Protocol (NCP) state display. In the LCP and NCP State display, verify the following items:

- In the PPP Link Control Protocol Status section of the display, verify under the LCP column that each path configured for Boundary Routing is in the OPEN state. The OPEN state indicates that both ends of the serial line connection are up and ready to bridge or route.
- In the PPP Network Control Protocol Status section of the display, verify in the BRIDGE column that the wide area port configured for Boundary Routing is in the OPEN state. The Network Control Protocol Status for all protocols other than bridging should be in the DISABLED state.

- 6 If you are operating the Boundary Routing feature over Frame Relay, verify the data link connection identifier (DLCI) status for all active Frame Relay ports by entering:

```
SHoW -FR DLciStat
```

Verify that the status of the link is active. If a DLCI is not in the list, the corresponding virtual port is down.

- 7 If you are operating the Boundary Routing feature over X.25, verify the status of the virtual circuits by entering:

```
SHoW -X25 STATUS
```

Verify that the state of the virtual circuits is up and running. Also verify that the DTE addresses and the protocols running on the virtual circuits are as you configured them.

- 8 If you are operating the Boundary Routing feature over an Synchronous Data Link Control (SDLC) line, verify the status of the central unit (CU) by entering:

```
SHoW -SDLC CUStatus
```

Troubleshooting the Configuration

If you are unable to make connections to the leaf network after configuring the central node, perform the following troubleshooting procedure. If your configuration continues to operate improperly, contact your network supplier or 3Com for assistance.

To troubleshoot the Boundary Routing configuration, follow these steps:

- 1 Make sure all cables on the central site network and the leaf network are properly connected.

For installation instructions, see the installation guides that shipped with the central and peripheral nodes.

- 2 Verify that the software configuration of the central node is correct.

For details, see "Verifying the Configuration" earlier in this chapter.

- 3 If you are managing the peripheral node from an SNMP agent, verify that the central node is configured to respond to incoming SNMP requests by entering:

```
SHoW -SNMP CONTroL
```

By default, the value of this parameter is set to Manage. If the value of this parameter is not Manage, enter:

```
SETDefault -SNMP CONTroL = Manage
```

- 4 In TCP/IP environments, make sure that you have correctly configured the default gateway.

Because the central node is the bridge/router in a Boundary Routing environment, use its MAC address (instead of the MAC address of the peripheral node) as the default gateway address when configuring clients on the leaf network that need access to hosts on the central site network.

- 5 If you see console messages that indicate smart filtering operations have stopped on a port, you can obtain information about the cause of the failure using:

```
SHoW !<port> -PORT DIAGnoStics
```

To troubleshoot the smart filtering configuration, follow these steps:

- a Verify that the link is up and stable.
If the link is prone to dropping packets, smart filtering operations will cease.
- b Verify that the Boundary Routing feature is configured correctly and operating.
For port, virtual port, and path configuration steps, see the Configuring Advanced Ports and Paths chapter. For Boundary Routing port configuration steps, see “Configuring Basic Boundary Routing” earlier in this chapter.
- c Make sure the peripheral node supports smart filtering.
If your peripheral node is running pre-7.0 Boundary Routing software, you must upgrade to software version 7.0 or later.
- d Make sure the size and configuration of your network is suitable for Boundary Routing and smart filtering operations.
For “out of memory” errors, you should try to decrease memory consumption. For example, you can use IPX policies to limit the view of the network. Optionally, you can increase memory or upgrade your peripheral node.

- e Restart smart filtering operations using:

```
SETDefault !<port> -PORT CONTROL = Enabled
```

- 6 Check the diagnostics of the central or peripheral node using:

```
SHow -DIAGnostic BoundaryCNDiag
```

The display obtained depends on the state of the router and how the router is being used.

- 7 If you are using the Boundary Routing feature in an IBM environment, verify that smart polling and data link switching (DLSw), if applicable, are functioning by following these steps:

- a Check the status of smart polling by entering:

```
SHow -SYS STATistics -LLC2
```

Verify that smart polling is functioning by comparing the number of RR frames received and transmitted by Boundary Routing and LAN ports. The number of RR frames received and transmitted by the Boundary Routing port should be substantially less than those on the LAN port.

Determine if Test and Xid frames are being received and transmitted on the correct ports.

For information on the LLC2 statistics display, see the Statistics Displays appendix.

- b Check the status of DLSw by entering:

```
SHow -DLSw Display
```

A display appears only if circuits are active or if an attempt to make a connection is being made.

- c Check the status of SDLC CUs by entering:

```
SHow -SDLC CUstatus
```

- 8 If you have configured the central node to perform the Boundary Routing operation of NetBIOS traffic and it does not appear to be performing this function, re-enable each port or virtual port on the central node using:

```
SETDefault !<port> -PORT CONTROL = Enabled
```

Customizing Boundary Routing

This section describes procedures you can use to customize your Boundary Routing configuration.

Configuring Dial-Related Enhancements

You can configure the following dial-related enhancements:

- Disaster recovery and bandwidth-on-demand in a PPP environment
- Disaster recovery in a Frame Relay environment
- Dial-on-demand in a PPP environment.

Beginning with software version 9.1, the bridge/router began using the concept of *bandwidth management*, a process that applies static bandwidth, dynamic bandwidth, or a combination of these to provide the ISDN and serial ports using PPP with the bandwidth they need to meet current requirements. Unlike versions of software previous to 9.1, bandwidth management does not view links as primary or secondary lines. It instead dynamically allocates or de-allocates unrestricted, available resources as needed to manage link traffic.

For information on configuring modems, see the *WAN Cabling and Connectivity Guide*. You can find this guide on the 3Com World Wide Web site by entering:

<http://www.3com.com/>

For conceptual information on configuring disaster recovery over Frame Relay, see the Configuring Wide Area Networking Using Frame Relay chapter.

Configuring Dual PVCs in a Boundary Routing Environment

This section describes how to configure a secondary permanent virtual circuit (PVC) dedicated to IBM traffic over a Frame Relay link in a Boundary Routing environment. The information in this section applies only to platforms that support the configuration of virtual ports. IBM traffic refers to both IBM System Network Architecture (SNA) and NetBIOS frames.

Dual PVCs are used in environments where IBM traffic is running with non-IBM traffic at a leaf node, and the IBM traffic is forwarded to a central site using Boundary Routing.

To implement dual PVCs, you configure two PVCs over a single Frame Relay Boundary Routing physical port (one PVC is dedicated to IBM traffic and the other PVC is dedicated to non-IBM traffic). Both PVCs are sent to a common bridge/router.

Configuration on the leaf nodes is completed by enabling the ports using the SETDefault -PORT CONTROL = Enable command after all other parameters have been set. In this particular configuration, enabling the ports triggers the transfer of the configuration information to the leaf nodes.

All configuration is performed on the central node.



For conceptual information on how dual PVCs work, see "Dual PVCs for IBM Traffic" later in this chapter.

Configuring Dual PVCs on the Central Node

The default condition for Frame Relay PVCs is that a single PVC is used to transmit all traffic types. To configure a separate PVC that will transmit only IBM traffic, you configure a virtual port *pair* on the Boundary Routing Frame Relay physical port on the central node. One of these virtual ports will then be configured to transmit non-IBM traffic. Doing this also indicates the virtual port to which IBM traffic should be redirected.

Figure 284 shows dual PVCs configured from a central node to two leaf nodes in an Ethernet environment. Figure 285 shows dual PVCs configured from a central node to two leaf nodes in a Token Ring environment.

Figure 284 Dual PVCs for IBM Traffic in a Boundary Routing Ethernet Environment

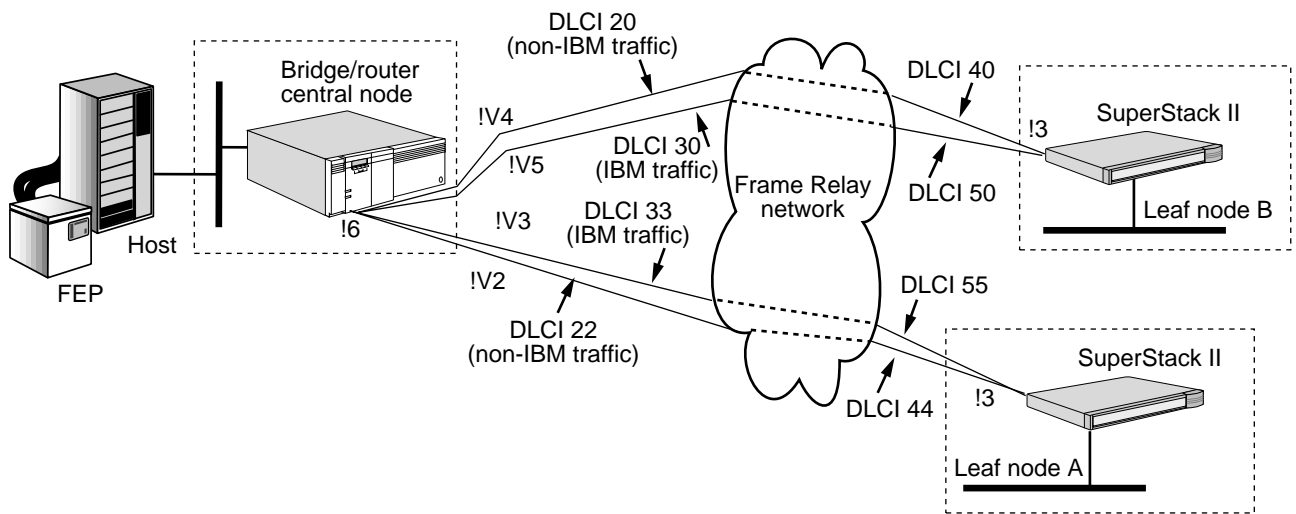
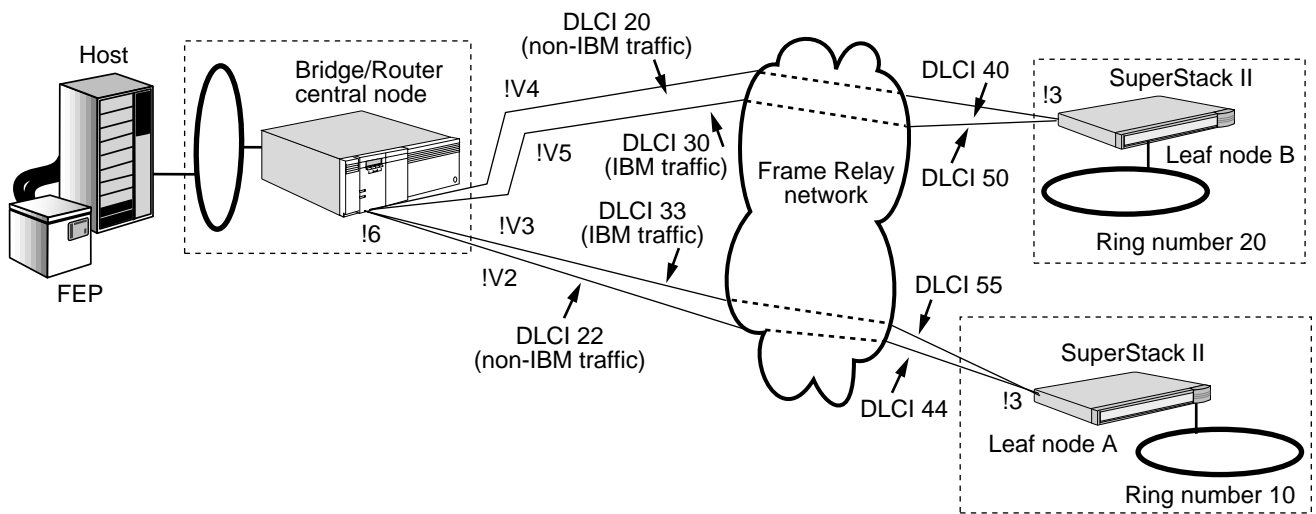


Figure 285 Dual PVCs for IBM Traffic in a Token Ring Environment



Prerequisites Before beginning these procedures, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your wide area interfaces.
- Acquire services from a Frame Relay service provider according to the *WAN Cabling and Connectivity Guide*. You can find this guide on the 3Com World Wide Web site by entering:

```
http://www.3com.com/
```

- Make sure ports that will be used for *non-IBM traffic* have the default DLCI value of 0. To check these values, use:

```
SHow !<Vport> -BCN LclNonIbmDlci
```

- Make sure virtual ports that will be used for *IBM traffic* have the -SR SrcRouBridge parameter configured to the default value SrcRouBridge.

See *Reference for Enterprise OS Software* for more information about this parameter and the default value of the DLCIs.

Basic Configuration for Both Ethernet and Token Ring To perform the basic configuration of dual PVCs on a central node, follow these steps:

- 1 Using Figure 284 or Figure 285 as an example, configure four virtual ports on Frame Relay physical port 6 by entering:

```
ADD !V2 -PORT VirtualPort 6@22
ADD !V3 -PORT VirtualPort 6@33
ADD !V4 -PORT VirtualPort 6@20
ADD !V5 -PORT VirtualPort 6@30
```

These commands configure two virtual ports to leaf node A (DLCI 22 and 33) and two to leaf node B (DLCI 20 and 30). The leaf nodes learn the DLCIs from the Frame Relay switch. Virtual ports 2 and 4 are for the PVCs carrying non-IBM traffic and virtual ports 3 and 5 are for the PVCs carrying IBM traffic.

- 2 Enable Boundary Routing of IBM traffic on virtual ports 3 and 5 by entering:

```
SETDefault !V3 -BCN CONTrol=IbmTraffic
SETDefault !V5 -BCN CONTrol=IbmTraffic
```

- 3 Define the PVC pairs by entering:

```
SETDefault !V3 -BCN LclNonIbmDlci=22
SETDefault !V5 -BCN LclNonIbmDlci=20
```

In Figure 284 and Figure 285, DLCIs 30 and 33 are the default IBM ports. The previous commands indicated to the central node that virtual ports 2 and 3 are one PVC pair, and virtual ports 4 and 5 are another PVC pair. They also instructed the central node to redirect IBM traffic to virtual ports 3 and 5, and to use virtual ports 2 (DLCI 22) and 4 (DLCI 20) for non-IBM traffic to the leaf nodes.

- 4 Instruct leaf node A to use DLCI 44 and leaf node B to use DLCI 40 for non-IBM traffic by entering:

```
SETDefault !V3 -BCN RemNonIbmDlci=44
SETDefault !V5 -BCN RemNonIbmDlci=40
```

- 5 Choose one of the following steps:
 - a If you are configuring dual PVCs in an Ethernet environment, proceed to “Enabling the Ports and Sending Leaf Nodes the Configuration Information” later in this chapter.
 - b If you are configuring dual PVCs in a Token Ring environment, continue with the procedure in the next section.

Additional Configuration Required for Token Ring Environments The following procedure enables source route bridging, defines IBM MAC addresses in non-canonical format, and configures Token Ring numbers on the virtual port.



The Token Ring port must be configured for transparent bridging when configuring a boundary router. This is the default setting, and must not be disabled.

Make sure you have completed steps 1 through 4 in “Basic Configuration for Both Ethernet and Token Ring” then continue with the following steps:

- 1 To configure Token Ring source route bridging on the ports illustrated in Figure 285, enter:

```
SETDefault !V2 -BCN RemoteLanType = TokenRing
SETDefault !V3 -BCN RemoteLanType = TokenRing
SETDefault !V4 -BCN RemoteLanType = TokenRing
SETDefault !V5 -BCN RemoteLanType = TokenRing
```

- 2 To specify IBM MAC addresses in non-canonical format, enter:

```
SETDefault !V2 -PORT PortMacAddrFmt = NonCanARP
SETDefault !V3 -PORT PortMacAddrFmt = NonCanARP
SETDefault !V4 -PORT PortMacAddrFmt = NonCanARP
SETDefault !V5 -PORT PortMacAddrFmt = NonCanARP
```

- 3 You must also configure the source route ring number and route discovery information on the IBM virtual ports. Assuming that the ring numbers for leaf node A and leaf node B are 10 and 20 respectively, enter:

```
SETDefault !V2 -SR RingNumber = None
SETDefault !V3 -SR RingNumber = 10
SETDefault !V4 -SR RingNumber = None
SETDefault !V5 -SR RingNumber = 20
SETDefault !V2 -SR RouteDiscovery = None
SETDefault !V3 -SR RouteDiscovery = LLC2
SETDefault !V4 -SR RouteDiscovery = None
SETDefault !V5 -SR RouteDiscovery = LLC2
```



The ring number and route discovery configuration to be used at the leaf node must be explicitly configured on the virtual ports running both IBM and non-IBM traffic.

Though ring numbers are mapped to the virtual ports, the ring numbers are actually used for the leaf node token ring LANs.

Proceed to the next section to complete this procedure.

Enabling the Ports and Sending Leaf Nodes the Configuration

Information This procedure enables the ports and transmits the needed configuration information to the leaf nodes. This is the last step for configuring dual PVCs in both Ethernet and Token Ring environments.

Enable the virtual ports, by entering:

```
SETDefault !V2 -PORT CONTROL = Enable
SETDefault !V3 -PORT CONTROL = Enable
SETDefault !V4 -PORT CONTROL = Enable
SETDefault !V5 -PORT CONTROL = Enable
```

Enabling these virtual ports triggers the transfer of the configuration information from the central node to the leaf nodes.

Verifying the Dual PVC Configuration

To verify that dual PVCs are correctly configured from the leaf node and the central node, follow these steps:

- 1 Verify that dual PVCs are configured from the leaf node using:

```
SHow [!<port> | !*] -FR DLciStat
```

This parameter displays the DLCI status and statistics for all active Frame Relay ports.

- 2 Verify that traffic is being transmitted through the ports using:

```
SHow -SYS STATistics -LLC2
```

- 3 Verify that dual PVCs are configured from the central node using:

```
SHow [!<port> | !*] -BCN IbmStatus
```

When the -BCN CONTROL parameter has been set to IbmTraffic, this parameter displays the status of Boundary Routing ports over which IBM traffic is running.

Configuring Network Resiliency

In a Boundary Routing topology, you can protect the operation of mission-critical applications if a failure occurs.



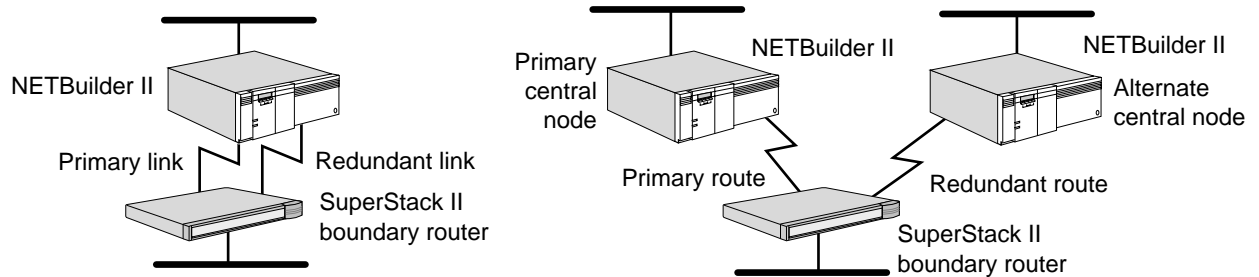
If you are configuring disaster recovery over Frame Relay on the peripheral node of a Boundary Routing environment, do not configure network resiliency.



If you are configuring network resiliency on a boundary router leaf network, make sure the PORT OWNER parameter and the PATH LineType parameter are set to values other than Auto. The port or path will not come up until you change those parameters to values other than Auto.

You can implement network resiliency in two different ways: you can configure a backup or *redundant link* between a central and peripheral node or a backup or *redundant route* to an alternate central node as shown in Figure 286.

Figure 286 Different Types of Network Resiliency



A redundant link provides a backup link if the primary link fails. A redundant route provides a backup route to an alternate central node if the primary route or the primary central node fails. The bandwidth management feature introduced in software version 9.1 views line resources as unrestricted, available resources, or resources configured for a specific function, such as disaster recovery only, instead of as primary and secondary lines.



When power is turned on, the auto startup feature brings up one active path only. It does not bring up a second path for network resiliency. You need to configure the second path for network resiliency using the procedure in the following section.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Determine how you want to implement network resiliency.

You must determine if you want to configure a redundant link between your central and peripheral nodes or if you want to configure a redundant route to an alternate central node. You must also determine which wide area networking protocol you will be using over the redundant link or route.
- See “Network Resiliency” later in this chapter to familiarize yourself with network resiliency and any configuration steps particular to your network resiliency implementation.
- Make sure you have fully configured the Boundary Routing port that you are planning the redundancy for (the primary link or route) according to instructions in “Configuring Basic Boundary Routing” earlier in this chapter.

Procedure

Use the following procedure to configure the redundant link or route in your Boundary Routing topology. The steps apply to both types of network resiliency configurations unless specifically called out. Complete the step on the central node unless specifically instructed to complete the step on both the central and peripheral nodes or the peripheral node only. See *Reference for Enterprise OS Software* for general information on parameters used in the following procedure.



Some of the steps in the following procedure require that you perform configuration on the peripheral node (boundary router). You can access the

Boundary Routing software through the System Configuration menu, an interface that prompts you to complete the tasks. If you cannot complete the steps outlined in this procedure using the System Configuration menu, you can exit the interface and access a command line user interface by selecting Quit from the System Configuration menu. To return to the menu, enter the InStall command. For more information on the Boundary Routing user interface, see the documentation that shipped with your boundary router.

To configure network resiliency, follow these steps:

- 1 Configure the wide area port and path according to instructions in the the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

If you are configuring a redundant link that will be running PPP, make certain that you assign the primary and secondary paths to one port using the -PORT PATHs parameter.

- 2 If you are using Frame Relay or X.25, create a virtual port for each leaf network that is attached to the Frame Relay or X.25 cloud according to instructions in the Configuring Wide Area Networking Using X.25 and Configuring Wide Area Networking Using Frame Relay chapters.
- 3 Configure the wide area or virtual port for the Boundary Routing feature according to “Configuring Basic Boundary Routing” earlier in this chapter.
- 4 Complete the following steps at both ends of the redundant link or route. See “Network Resiliency” later in this chapter for information on parameter settings for specific network resiliency configurations.

- a Assign an owner to the wide area port using:

```
SETDefault !<port> -PORT OWNEr = PPP | FrameRelay | X25
```

- b Set the line type on the wide area path using:

```
SETDefault !<path> -PATH LineType = Leased | Dialup | Auto
```

- c Set the attributes of the wide area path using:

```
SETDefault !<path> -PATH DialCONTRol = (UnReSTRicted |
DisasterRecovery | NoDisasterRecovery, [Answer [NoAnswer],
[Originate | NoOriginate])
```

- 5 If you are configuring a redundant PPP link as a backup to a primary PPP link, follow these steps:

- a Enable bandwidth-on-demand on both ends of the wide area link using:

```
SETDefault !<port> -PORT DialInitState = (DialonDemand)
```

- a Enable disaster recovery on both ends of the wide area link using:

```
SETDefault !<port> -PORT DialCONTRol = (DisasterRcvry)
```

- b If in an IBM Boundary Routing topology, set the value of the -LLC2 RetryCount to 20 on the Boundary Routing port using:

```
SETDefault !<port> -LLC2 RetryCount = 20
```

Increasing the value of this parameter from its default setting to 20 ensures that the retry timer will not time out and bring the circuit down, if the primary link goes down and the secondary line comes up.

- 6 If you are configuring a redundant PPP link as a backup to a primary Frame Relay link, configure the same address or network number (for example, an IP address or an IPX network number) on both wide area ports of the central node.

For more information on how to do this, see “Frame Relay Environment” and “IBM Environment” later in this chapter.

- 7 If you are configuring a redundant route to an alternate central node, follow these steps:

- a Enable automatic dialing at the peripheral node end of the redundant route only, using:

```
SETDefault !<port> -PORT AutoDial = Enabled
```

Setting the value of this parameter to Enabled allows the peripheral node to automatically dial the alternate central node if the primary route or primary central node fails.

- b Configure the same address or network number (for example, an IP address or an IPX network number) on the wide area ports of both central nodes.

See the appropriate bridging or routing chapter, for example, the chapter on IPX routing, for information on configuring an address or network number. For more information on why you need to complete this step, see “Primary and Alternate Central Node Configuration” later in this chapter.

- c If you plan to bridge or route AppleTalk, IP, or IPX in your Boundary Routing topology, enable the central MAC address on the wide area ports of both central nodes, using:

```
SETDefault !<port> -BCN CONTROL = CentralMac
```

You may also need to configure the wide area port on the alternate central node with the same bridging or routing attributes as the primary central node. To make this determination, see “Using the Central MAC Address” later in this chapter. If you determine that you need to do this, see the appropriate bridging or routing chapter, for example, the chapter on IPX routing, for information on configuring these attributes.

- d Disable the wide area port on the alternate central node using:

```
SETDefault !<port> -PORT CONTROL = Disabled
```

Disabling this port prevents it from coming up before the primary central node port. It also can control which central node is primary and which is alternate instead of the software negotiation making the decision.

If the primary route or primary central node fails, you can enable the wide area port on the alternate central node using:

```
SETDefault !<port> -PORT CONTROL = Enabled
```

- e If you are configuring a redundant PPP route as a backup to a primary X.25 route, you must make certain that if the X.25 route goes down at the peripheral node, the virtual port associated with this route is disabled at the central node before the redundant PPP route activates.

For more information on how to do this, see “X.25 Environment” and “IBM Environment” later in this chapter.

How Boundary Routing System Architecture Works

Boundary Routing system architecture can interconnect networks using wide area links. Boundary Routing is ideally suited for environments that require a large number of small remote office networks (leaf networks) to be connected to a central office (central site network).

Where Can Boundary Routing Be Used?

To implement Boundary Routing system architecture in a network topology, a central node (which provides the routing function) and peripheral nodes must be present.

The central node must be a NETBuilder II bridge/router, or a model 227, 327, 427, 527 SuperStack II NETBuilder bridge/router, or a model 147 OfficeConnect NETBuilder bridge/router. The central node can be configured as a bridge or a router.

Table 75 describes the platforms that can be used as a central node.

Table 75 Central Node Information

| Platform | Number of Peripheral Nodes Supported | Protocols Supported |
|---|--|---|
| NETBuilder II bridge/router | Up to 40 or 75 peripheral nodes (software packages SW/NBII-CP and SW/NBII-FF support 75 peripheral nodes). No restrictions as to the number of peripheral nodes the NETBuilder II bridge/router supports over PPP, Frame Relay, or X.25. | Bridging and all routing protocols |
| Model 227*
SuperStack II bridge/router | Up to 10 peripheral nodes over Frame Relay.
Up to 3 peripheral nodes using PPP over 56/64 KB leased or dial-up lines.
Must use a SuperStack II NETBuilder boundary router that supports Ethernet as the LAN media type as a peripheral node. | Bridging and all routing protocols except APPN |
| Model 327*
SuperStack II bridge/router | Up to 10 peripheral nodes over Frame Relay.
Up to 3 peripheral nodes over PPP.
Must use a SuperStack II NETBuilder boundary router that supports token ring as the LAN media as a peripheral node. | Source route bridging and all routing protocols except APPN. Does not support transparent bridging. |
| Model 427*
SuperStack II bridge/router and Model 147 OfficeConnect bridge/router | Up to 10 peripheral nodes over Frame Relay.
Up to 3 peripheral nodes using PPP over 56/64 KB leased or dial-up lines.
Must use a SuperStack II NETBuilder boundary router that supports Ethernet as the LAN media type as a peripheral node. | Bridging and all routing protocols except APPN |
| Model 527*
SuperStack II bridge/router | Up to 10 peripheral nodes over Frame Relay.
Up to 3 peripheral nodes over PPP.
Must use a SuperStack II NETBuilder boundary router that supports token ring as the LAN media as a peripheral node. | Source route bridging and all routing protocols except APPN. Does not support transparent bridging. |

* These platforms do not support the IBM network management application LAN Net Manager (LNM).

The peripheral node can be any 3Com platform that runs the Boundary Routing system architecture software. For example, the peripheral node can be one of the following devices:

- SuperStack II and OfficeConnect NETBuilder boundary router
- LinkBuilder® Ether Connect System (ECS) Remote Control Module (runs Boundary Routing over PPP only)

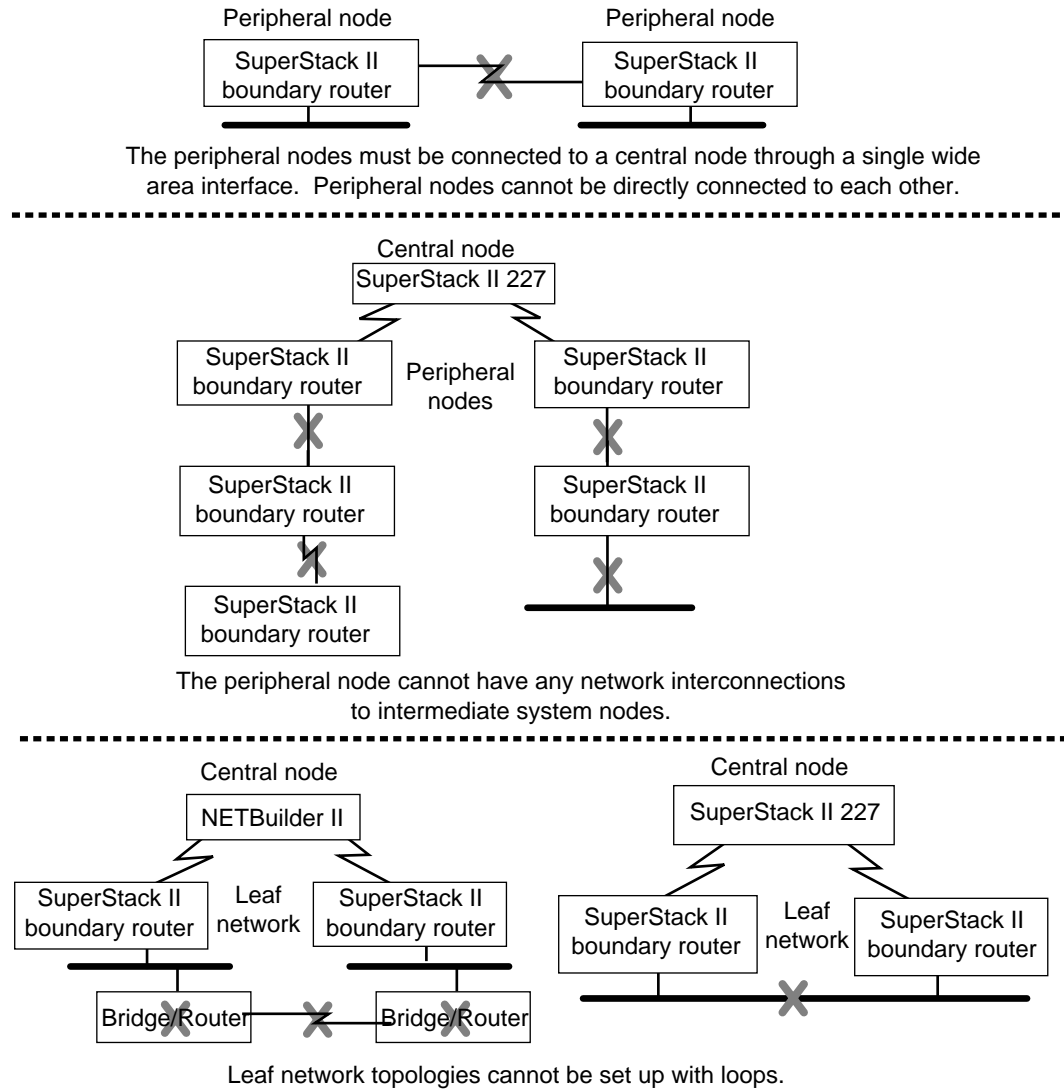
In Boundary Routing network topologies, the following rules apply:

- The remote office networks must be leaf networks. A single (one and only one) active network interconnection from the leaf network to the central node is permitted.
- The peripheral node and the central node must be connected over a point-to-point serial link or a virtual circuit.
- The central node is the bridge/router in a Boundary Routing environment. Any client configuration on the leaf network that requires addressing the router needs to use the address information pertaining to the central node, not the peripheral node.

A backup link can be configured for bandwidth-on-demand, dial-on-demand, or network resiliency. A backup route to an alternate central node can be configured for network resiliency.

Figure 287 shows illegal topologies in which Boundary Routing cannot operate.

Figure 287 Illegal Boundary Routing Topologies



Typical Boundary Routing Environment

This section provides examples of the following types of Boundary Routing environments:

- Non-IBM
 - Using a NETBuilder II bridge/router as a central node
 - Using a model 227 or 427 SuperStack II NETBuilder bridge/router as a central node
- IBM
 - Using a NETBuilder II bridge/router as a central node
 - Using a NETBuilder II bridge/router as a regional central node
 - Using a model 327 or 527 SuperStack II NETBuilder bridge/router as a central node
 - APPN

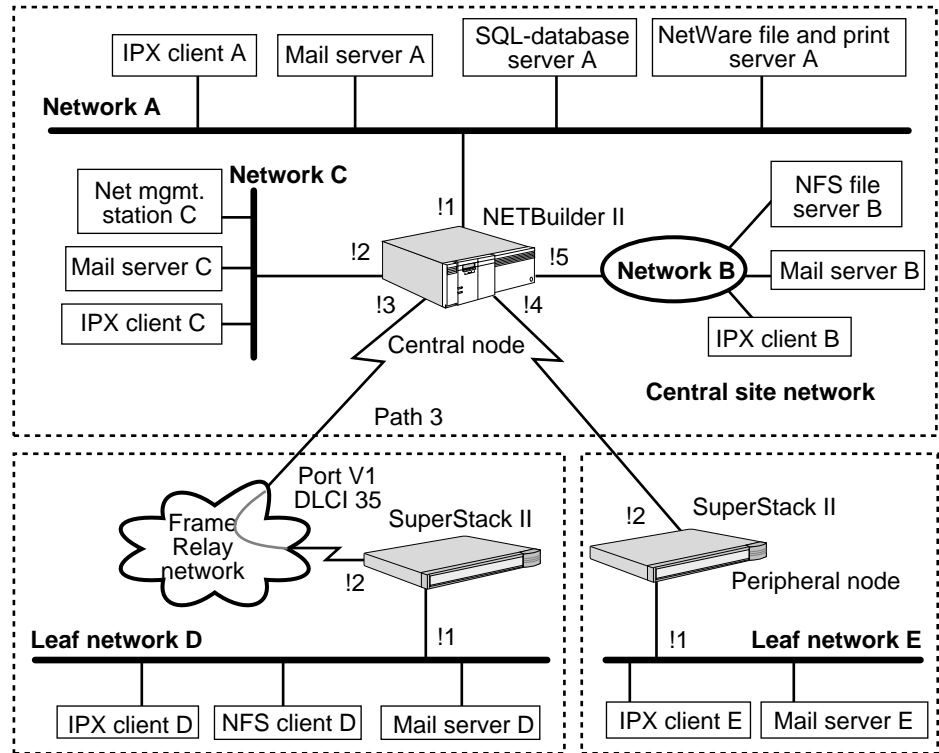


BSC traffic is not supported in Boundary Routing environments.

Non-IBM Environment Using a NETBuilder II Bridge/Router

Figure 288 shows a typical non-IBM environment using Boundary Routing system architecture. A NETBuilder II bridge/router is used as the central node.

Figure 288 Typical Non-IBM Boundary Routing Environment Using NETBuilder II



In this figure, the central site network comprises networks A, B, and C, and two leaf networks D and E. The leaf networks are connected to the central site network using Boundary Routing system architecture.

Network A contains a mail server for electronic mail exchange, a Structured Query Language (SQL) database server, and a NetWare file and print server. All hosts on this network use IPX as the underlying network protocol.

Network B is a multiprotocol token ring network containing a Network File System (NFS) file server and a mail server for electronic mail exchange. The file server uses IP as the underlying network protocol, and the mail server uses IPX.

Network C is also a multiprotocol network containing a mail server and a network management station. The mail server uses IPX as the underlying protocol; the network management station provides SNMP and Telnet, both of which use IP as the underlying protocol. The network management station manages the central node and the peripheral nodes.

Leaf network D, which uses a virtual port and is connected across a Frame Relay network, requires access to the NetWare file and print server, the SQL database server, and the NFS file server. Leaf network D also exchanges electronic mail with the central site network and leaf network E. NetWare, SQL, and electronic mail are run over IPX. NFS is run over IP.

Leaf network E requires access to the NetWare file and print server, and exchanges electronic mail with the central site network and leaf network D. NetWare and electronic mail are run over IPX.

If you use Boundary Routing system architecture to interconnect these networks, the following configuration steps are required:

- On the central site:
 - Configure the WAN links.
 - Enable Boundary Routing on port 4 and virtual port V1.
 - Configure RARP to assign IP addresses to the peripheral nodes.
 - Assign IP network addresses for networks B, C, and D.
 - Assign IPX network addresses for networks A, B, C, D, and E.
 - Configure ports 1, 2, 4, and 5 and virtual port V1 of the central node to route IPX.
 - Configure ports 2 and 5 and virtual port V1 of the central node to route IP.
 - Configure the remote LAN type on port 4 and virtual port V1.
- On the peripheral nodes:

In most cases, no configuration is necessary on the peripheral node. See the documentation that accompanies your peripheral node to determine if configuration is necessary.

When using Boundary Routing system architecture to achieve connectivity, fewer configuration steps are required at each leaf network, resulting in administrative savings and troubleshooting costs if the number of leaf networks is large.

Non-IBM Environment Using a SuperStack II Bridge/Router Model 227 or 427

Figure 289 and Figure 290 show typical non-IBM environments in which Boundary Routing system architecture is used. In Figure 289 a model 227 SuperStack II bridge/router is the central node, while in Figure 290 model 427 SuperStack II bridge/router is the central node.

Figure 289 Model 2267 Bridge/Router in a Typical Non-IBM Boundary Routing Environment

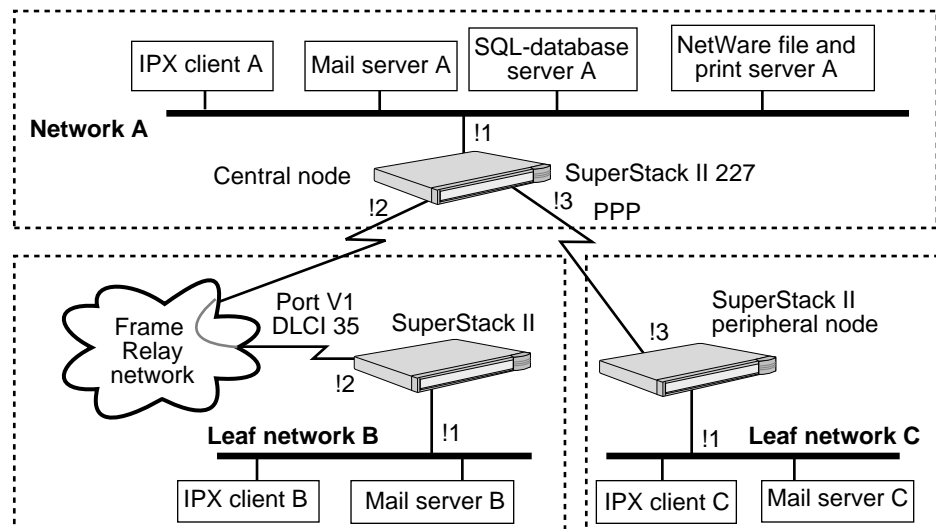
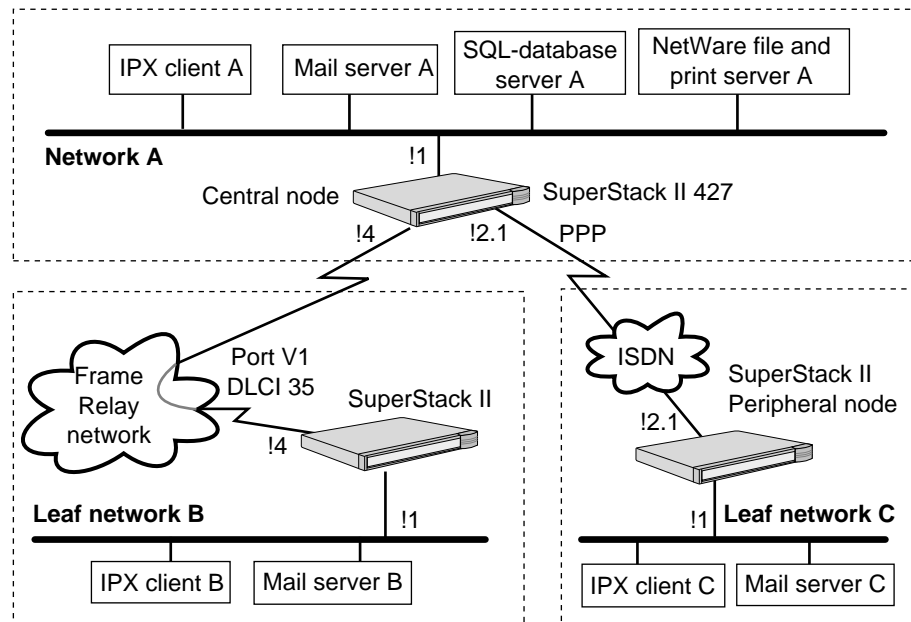


Figure 290 Model 427 Bridge/Router in a Typical Non-IBM Boundary Routing

Environment

For information on the number of peripheral nodes supported by model 227 and model 427 SuperStack II bridge/routers, see Table 75.

In these figures, the central site network is network A and the leaf networks are B and C. The leaf networks are connected to the central site network using Boundary Routing system architecture. IPX is the underlying network protocol in these topologies.

Network A contains a mail server for electronic mail exchange, an SQL database server, and a NetWare file and print server.

Leaf network B, which uses a virtual port and is connected across a Frame Relay network, requires access to the NetWare file and print server and the SQL database server. Leaf network B also exchanges electronic mail with the central site network and leaf network C.

In Figure 289, network C is connected across a PPP network while in Figure 290, it is connected across an ISDN network. Leaf network C requires access to the NetWare file and print server, and exchanges electronic mail with the central site network and leaf network B.

If you use Boundary Routing system architecture to interconnect these networks, the following configuration steps are required:

- On the central site:
 - Configure the WAN links, including the ISDN interface on model 427 bridge/router.
 - Enable the Boundary Routing feature on ports V1 and 3 in Figure 289 and ports V1 and 2.1 in Figure 290.
 - Assign IPX network addresses for networks A, B, and C.

- Configure ports 1, 3, and V1 in Figure 289 and ports 1, 2.1, V1 in Figure 290 of the central node to route IPX.
- Configure the remote LAN type on ports V1 and 3 in Figure 289 and ports V1 and 2.1 in Figure 290.
- On the peripheral nodes:
In most cases, no configuration is necessary on the peripheral node. See the documentation that accompanies your peripheral node to determine if configuration is necessary.

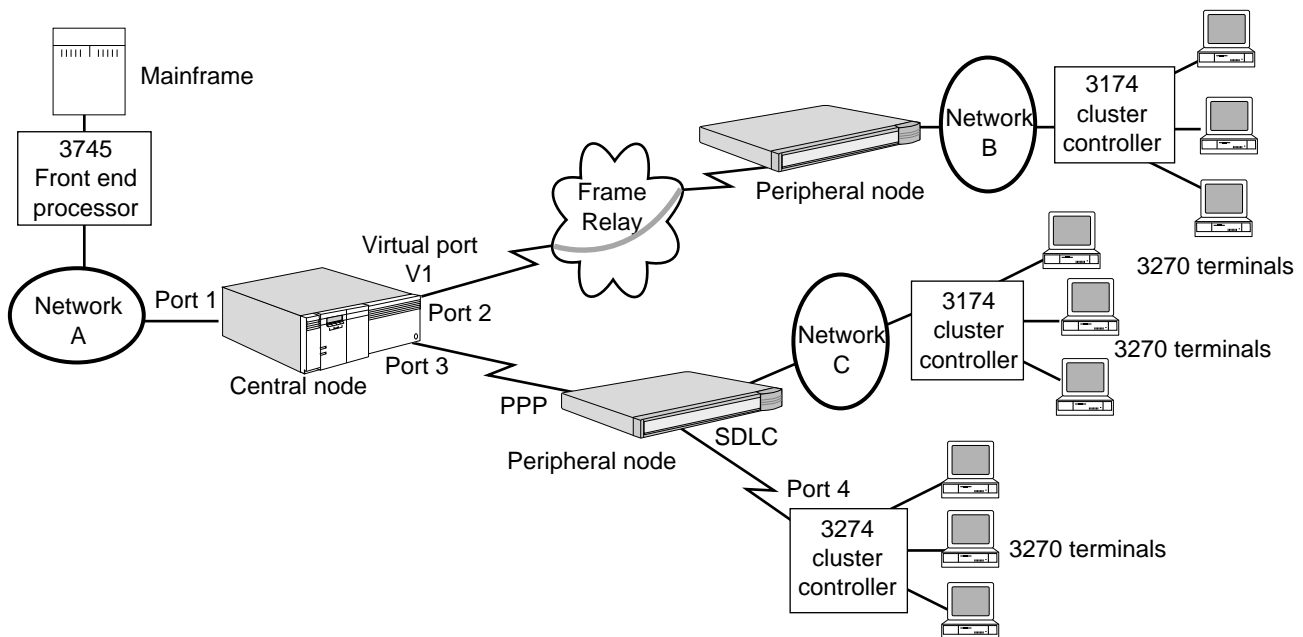
When using Boundary Routing system architecture to achieve connectivity, fewer configuration steps are required at each leaf network, resulting in administrative and troubleshooting cost savings if the number of leaf networks is large.

IBM Environment Using a NETBuilder II Bridge/Router as a Central Node

Although an SNA example is used, the information in this section applies to both SNA and NetBIOS topologies except where specifically noted.

Figure 291 shows an SNA Boundary Routing topology with a NETBuilder II bridge/router as a central node.

Figure 291 SNA Boundary Routing Topology: NETBuilder II As Central Node



In this figure, the central site network is network A and the leaf networks are networks B and C. The leaf networks are connected to the central site network using Boundary Routing system architecture.

Network A contains an IBM 3745 front-end processor (FEP) and a mainframe computer. Networks B and C contain remote 3174 cluster controllers and 3270 terminals, which must periodically access applications on the mainframe computer using SNA LLC2 sessions or SDLC sessions.

If you use Boundary Routing system architecture to interconnect these networks, the following configuration steps are required:

- On the central site:
 - For NetBIOS topologies, enable Boundary Routing of NetBIOS traffic.
 - Configure the LLC2 data link interface on port 1.
 - Enable source route bridging on port 1.
 - Assign unique ring number on port 1.
 - Configure the WAN links.
 - Enable Boundary Routing on port 3 and virtual port V1.
 - Configure the remote LAN type on port 3 and virtual port V1.
 - Set the ARP address format to noncanonical for port 3 and virtual port V1.
 - Enable Boundary Routing of IBM traffic on port 3 and virtual port V1.
- On the peripheral nodes:
 - In most cases, no configuration is necessary on the peripheral node. See the documentation that accompanies your peripheral node to determine if configuration is necessary.
 - If you are configuring SDLC as a client protocol over the boundary router link, you must configure the SDLC port and path attributes, as well as the CU information, on the peripheral node. For more information, see the *Configuring Synchronous Data Link Control Connectivity* chapter.
For sending SDLC traffic over a boundary router link, no additional configuration is required at the central node.

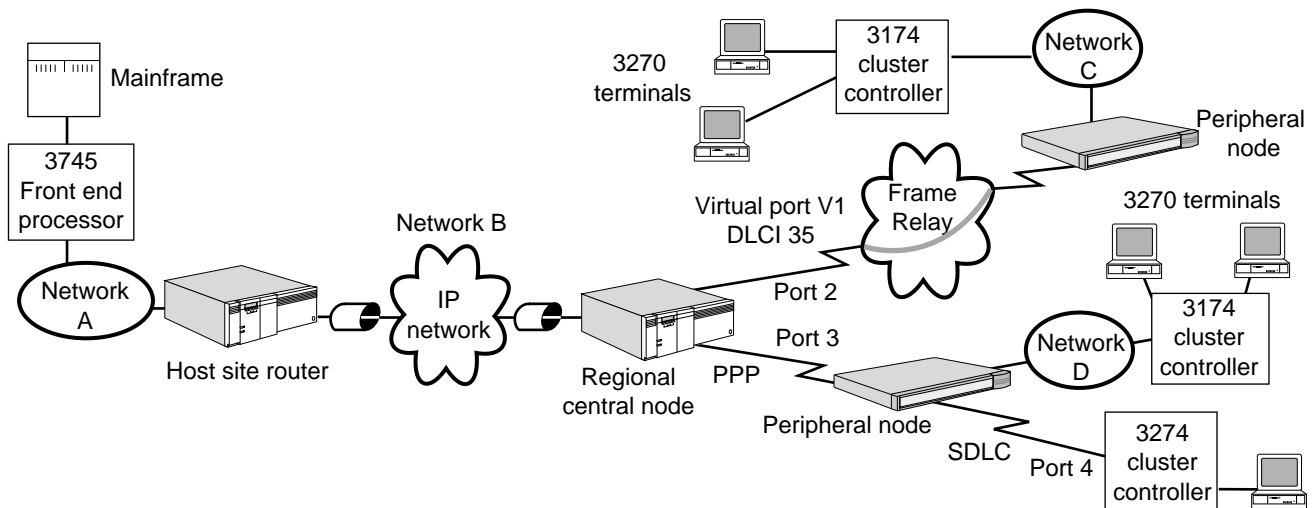
When using Boundary Routing system architecture to achieve connectivity, fewer configuration steps are required at each leaf network, resulting in administrative and troubleshooting cost savings if the number of leaf networks is large.

IBM Environment Using a NETBuilder II Bridge/Router as a Regional Central Node

Although an SNA example is used, the information in this section applies to both SNA and NetBIOS topologies except where specifically noted.

Figure 292 shows Boundary Routing system architecture in an SNA environment with a NETBuilder II bridge/router as a regional central node.

Figure 292 SNA Boundary Routing Topology: NETBuilder II As Regional Central Node



In this figure, the central site networks are networks A and B and the leaf networks are networks C and D. The leaf networks are connected to the central site network using Boundary Routing system architecture.

Network A contains an IBM 3745 FEP and a mainframe computer. Networks C and D contain remote 3174 cluster controllers and 3270 terminals, which must periodically access applications on the mainframe computer using SNA LLC2 sessions or SDLC sessions.

This topology differs from the traditional Boundary Routing topology because packets that passed between the mainframe and the terminals and vice versa must additionally traverse network B, which is an IP internetwork. DLSw enabled on both the regional central node and the host site router allows the SNA traffic to traverse the IP internetwork. For more information on DLSw, see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.

If you use Boundary Routing system architecture to interconnect these networks, the following configuration steps are required:

- On the host site:
 - For NetBIOS topologies, enable Boundary Routing of NetBIOS traffic.
 - Configure the LLC2 data link interface on LAN port.
 - Enable source route bridging on LAN port.
 - Assign unique ring number on LAN port.
 - Configure DLSw on WAN port.
- On the regional central site:

- Configure the WAN links.
- Enable Boundary Routing on port 3 and virtual port V1.
- Configure the remote LAN type on port 3 and virtual port V1.
- Set the ARP address format to noncanonical for port 3 and virtual port V1.
- Enable Boundary Routing of IBM traffic on port 3 and virtual port V1.
- Configure DLSw on the port that interfaces network B, the IP internetwork.
- On the peripheral nodes:
 - In most cases, no configuration is necessary on the peripheral node. See the documentation that accompanies your peripheral node to determine if configuration is necessary.
 - If you are configuring SDLC as a client protocol over the boundary router link, you must configure the SDLC port and path attributes and the CU information on the peripheral node. For more information, see the *Configuring Synchronous Data Link Control Connectivity* chapter.
For sending SDLC traffic over a boundary router link, no additional configuration is required at the central node.

When using Boundary Routing system architecture to achieve connectivity, fewer configuration steps are required at each leaf network, resulting in administrative and troubleshooting cost savings if the number of leaf networks is large.

IBM Environment Using a SuperStack II NETBuilder Bridge/Router Model 327 or 527 As a Central Node

Figure 293 and Figure 294 show SNA Boundary Routing topologies with model 327 and 527 SuperStack II NETBuilder bridge/routers as central nodes.

Figure 293 SNA Boundary Routing Topology: Model 327 SuperStack II Central Node

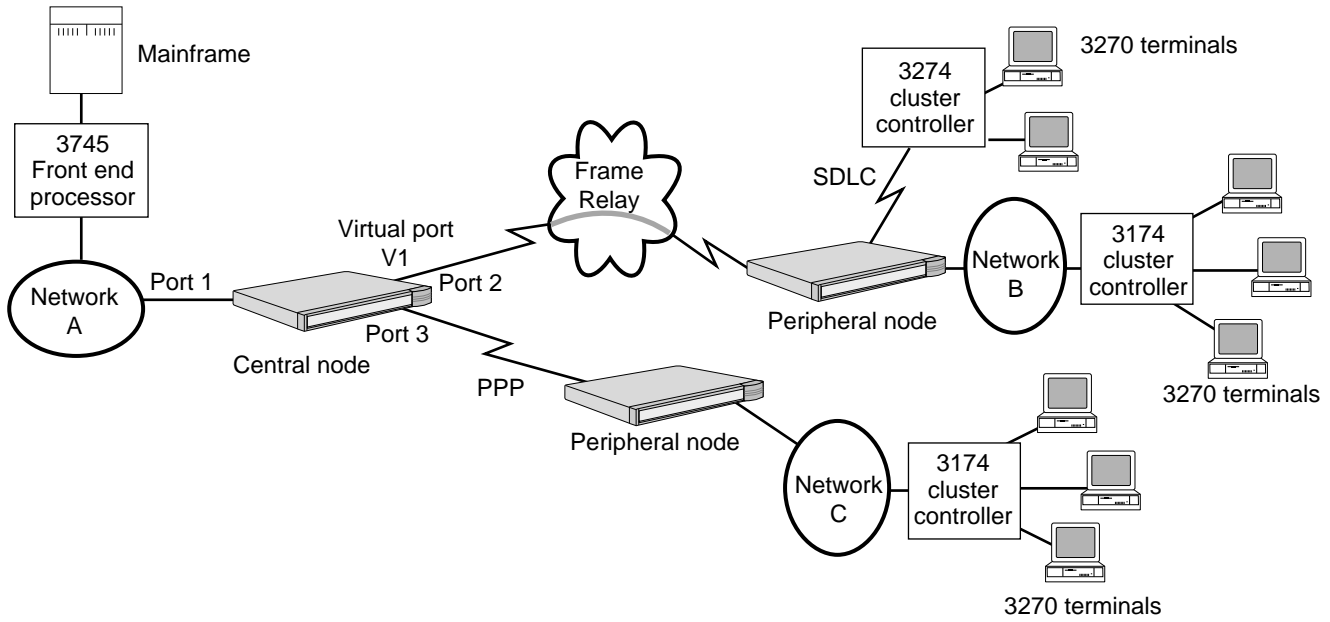
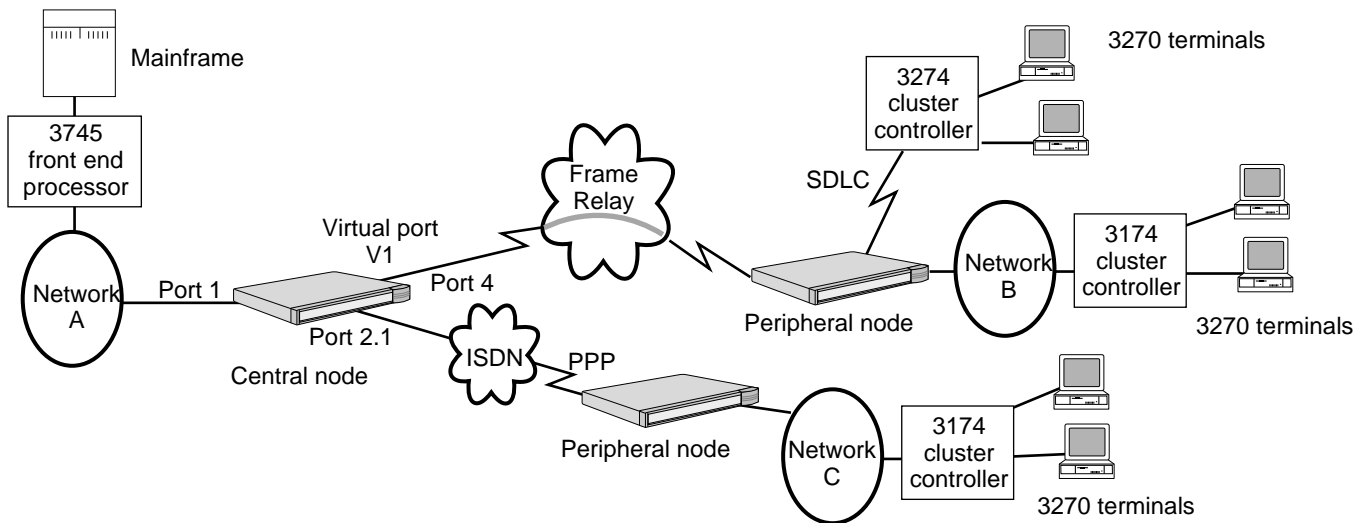


Figure 294 SNA Boundary Routing Topology: Model 527 SuperStack II Central Node



In both figures, the central site network is network A and the leaf networks are networks B and C. The leaf networks are connected to the central site network using Boundary Routing system architecture.

Network A contains an IBM 3745 FEP and a mainframe computer. Networks B and C contain remote 3174 cluster controllers and 3270 terminals, which must periodically access applications on the mainframe computer using SNA LLC2 sessions. The boundary router that is connected to network B also provides a connection to an SDLC device, in this case, a remote 3274 cluster controller with 3270 terminals attached. The terminals must periodically access applications on the mainframe computer using SDLC sessions.

If you use Boundary Routing system architecture to interconnect these networks, the following configuration steps are required:

- On the central site:
 - Configure the LLC2 data link interface on port 1.
 - Enable source route bridging on port 1.
 - Assign unique ring number on port 1.
 - Configure the WAN links, including the ISDN interface on model 527.
 - Enable Boundary Routing on ports V1 and 3 in Figure 293 and on ports V1 and 2.1 in Figure 294.
 - Configure the remote LAN type on ports V1 and 3 in Figure 293 and on ports V1 and 2.1 in Figure 294.
 - Set the address format for ARP to noncanonical on ports V1 and 3 in Figure 293 and on ports V1 and 2.1 in Figure 294.
 - Enable Boundary Routing of IBM traffic on ports V1 and 3 in Figure 293 and on ports V1 and 2.1 in Figure 294.
- On the peripheral nodes:
 - In most cases, no configuration is necessary on the peripheral node. See the documentation that accompanies your peripheral node to determine if configuration is necessary.
 - If you are configuring SDLC as a client protocol over the boundary router link, you must configure the SDLC port and path attributes and CU information on the peripheral node. For more information, see the *Configuring Synchronous Data Link Control Connectivity* chapter.
For sending SDLC traffic over a boundary router link, no additional configuration is required at the central node.

When using Boundary Routing system architecture to achieve connectivity, fewer configuration steps are required at each leaf network, resulting in administrative savings and troubleshooting costs if the number of leaf networks is large.

APPN Topology

For more information on Boundary Routing in an APPN environment, see the *APPN High Performance Routing* chapter.

SDLC Over Boundary Router Links

You can attach an SNA/SDLC system to a peripheral node, and send SDLC traffic over a boundary router link. To do this, no additional configuration of WAN ports at the central node is required. All SDLC configuration required for routing SDLC traffic over a boundary router link is performed at the peripheral node.

If you plan to configure SDLC on a SuperStack II bridge/router, see the appropriate SuperStack II Ethernet or Token Ring guide.

Boundary Routing Features

The Boundary Routing software provides the following advantages when connecting remote office networks:

- Simplifies network administration through configuration at the central node.
- Reduces WAN usage costs through smart filtering, dial-on-demand, payload or data compression, and data exchange with specific peers in an IBM Boundary Routing topology.
- Provides higher reliability through local termination and the automatic prioritization of IBM traffic in an IBM Boundary Routing topology.
- Provides continuous operation with a dial-up backup line for disaster recovery and bandwidth-on-demand, and a mechanism for constructing resilient networks.

Simplified Network Administration

In remote office network environments, Boundary Routing system architecture can be used to construct a manageable network topology and simplify network administration. The topology is manageable because routing is used to switch packets between the leaf networks and the central site network. This allows for greater flexibility in network segmentation and better control over the traffic. Administration is simplified because, unlike traditional routing where the administrative burden is on both ends of the interconnection, most of the administration is performed at the central site network. A few, simple configurations may be needed at the leaf networks. The leaf networks often may require no configuration at all.

Reduced WAN Usage Costs

The following features reduce WAN usage costs.

Smart Filtering Smart filtering reduces the cost associated with WAN lines by minimizing the number of packets that must be sent over the WAN link, particularly overhead traffic such as topology-maintenance messages. This feature is called smart filtering because the filtering decisions are automatically made by the central node in the Boundary Routing system based on the configuration at the central site and the traffic flow from the remote device. The filtering actions are then taken by the peripheral node.

You can use smart filtering in an IPX environment and an extension of smart filtering in an IBM environment called *smart polling*.

You can use smart filtering to do the following if you are using the IPX Protocol in your Boundary Routing topology:

- Eliminate non-IBM traffic belonging to protocol islands that are confined to a leaf network from the WAN link when the central node is strictly routing on the Boundary Routing port (bridging has been disabled).

As shown in Figure 295, the VINES, AppleTalk, and LAT clouds represent protocol islands and have *no* connection needs with other leaf networks or the central node. Protocol islands consist of network topologies that are always confined to a single leaf network and have no interconnection needs with other leaf networks or the central node. Smart filtering prevents traffic generated by these protocol islands from being forwarded over the WAN link because the central node instructs the peripheral node to filter.

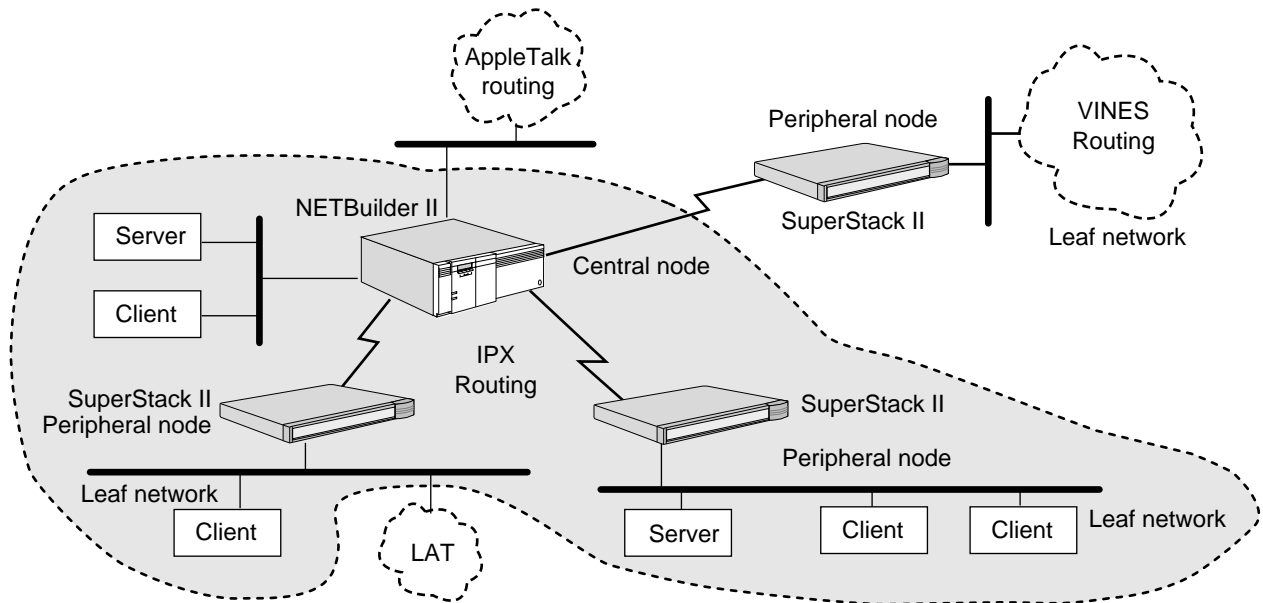


If bridging is enabled on the Boundary Routing port, all traffic, including protocol island traffic, is forwarded.

- Eliminate periodic rebroadcasts of IPX NRIP and SAP updates from the central node and NetWare servers on the leaf networks on the WAN link without requiring static configurations of routes or services at either end.

To enable smart filtering, set the value of the -BCN CONTROL parameter to SmartFiltering.

Figure 295 Protocol Islands in Boundary Routing Environments



The extension of smart filtering for IBM topologies called smart polling is available in software version 8.2 and later.

SNA and NetBIOS use the data link protocol LLC2. After a user at a terminal initiates an LLC2 session with an SNA or NetBIOS host, polling packets are exchanged continually between the central node and the peripheral node during the session to indicate that the LLC2 session is still alive. If multiple LLC2 sessions

between the host and clients are running, the number of polling packets exchanged by the central and peripheral nodes becomes significant.

Smart polling reduces the number of polling packets exchanged between the central and peripheral nodes. For example, suppose that four LLC2 sessions are running simultaneously between a mainframe computer and four different terminals on the same leaf network. Instead of the central and peripheral nodes exchanging polling packets for each session, the central and peripheral nodes assume that a poll reply for one session indicates that the other three sessions are still alive.

Smart polling is effective when one or multiple LLC2 sessions are running simultaneously between a host and terminals on the same leaf network. In fact, the more sessions that are running simultaneously, the greater the reduction of the number of polling packets sent over the WAN link.

To activate smart polling, set the value of the `-BCN CONTROL` parameter to `lbmTraffic`.

Smart Filtering for Boundary Routing over X.25 If smart filtering is operating on a peripheral node, and the link between the peripheral node and the X.25 packet-switched network is inoperable, then the central node and the peripheral node can become unsynchronized. The result is that the NetWare servers at the remote site are not refreshed with information about other NetWare servers located at the central site. This condition can be corrected by re-enabling the affected virtual port on the central site router.

Smart Filtering and SAP The smart filtering feature for ports using Boundary Routing software cannot be used when the size of the SAP table in your network exceeds 400 services.

Before enabling smart filtering, check the size of your SAP information table by entering:

```
SHow -IPX AllServers
```

If the number displayed is greater than 400, do not attempt to enable smart filtering on any of the ports that use Boundary Routing unless you use the `Advertise Policy` parameter to control the list of SAP entries.

If there are only NetWare clients at the remote site and no NetWare servers, another way to reduce SAP traffic over ports using Boundary Routing is to turn SAP talk off by using the `CONTROL` parameter in the `SAP Service` on the WAN ports at the central site. Leave the SAP talk on LAN ports. To turn SAP talk off, use:

```
SETDefault !<port> -SAP CONTROL = NoTalk
```

If the remote site has a server, you can run NLSP on the Boundary Routing port between the router and the server. NLSP is supported on version 8.0 and higher with Netware 3.12, 4.01, and 4.1.

Smart Filtering in a Boundary Routing Topology If you have enabled the smart filtering feature in your Boundary Routing topology and have subsequently added or deleted IPX services on a server on a currently active leaf network without restarting the server, you must disable then re-enable smart filtering on the central node. Re-enabling smart filtering on the central node enables it to update the services learned from the remote leaf network.

To disable smart filtering, use:

```
SETDefault !<port> -BCN CONTrol = NoSmartFiltering
```

To enable smart filtering, use:

```
SETDefault !<port> -BCN CONTrol = SmartFiltering
```

Disabling Smart Filtering If you want to disable the smart filtering feature and have enabled the smart filtering feature in your Boundary Routing topology, you should re-enable the port after the smart filtering feature is disabled. To re-enable the port after smart filtering is disabled, use:

```
SETDefault !<port> -PORT CONTrol = Disabled
```

```
SETDefault !<port> -PORT CONTrol = Enabled
```

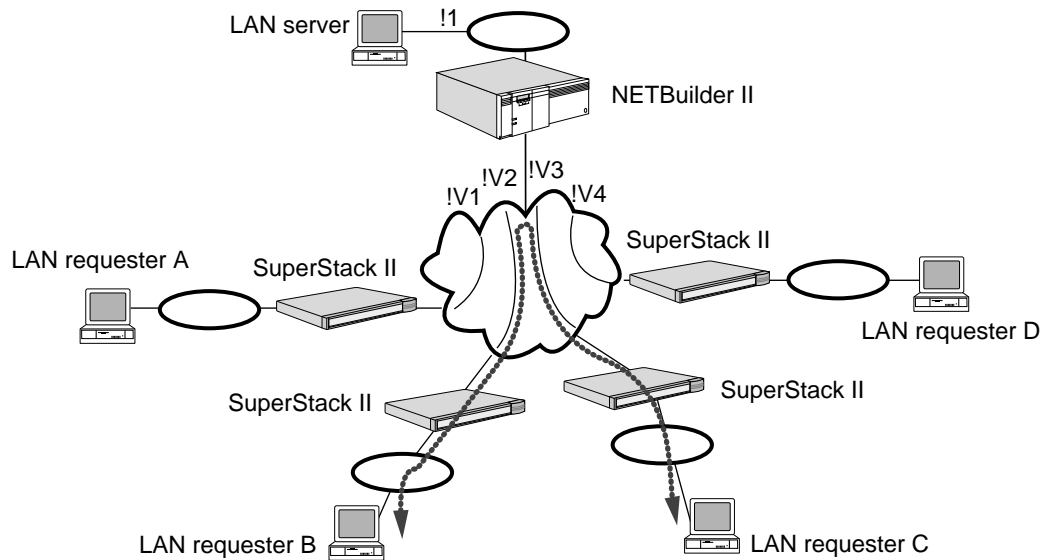
Dial-On-Demand To further reduce phone line costs when communicating over a WAN link in a Boundary Routing environment, you can configure the WAN link to be a dial-on-demand (DOD) line. When a demand occurs (user data needs to be transmitted), DOD automatically makes the call to establish the connection. The call is then terminated and reestablished automatically without any intervention depending upon whether or not there is data to be sent across the line. Connections that are no longer in use are temporarily terminated until a new demand occurs.

When routing IPX over a DOD line in a Boundary Routing environment, you can use the IPX spoofing feature in software version 8.0 and later to control the number of NetWare Communication Protocols (NCP) KeepAliveRequest packets (also known as WatchDog packets) from central node servers to the peripheral node clients. Spoofing helps manage the amount of traffic over the DOD line without violating the integrity of NCP connection maintenance.

Data Compression You can use data compression in all types of Boundary Routing topologies, but in particular, using the data compression feature in an SNA Boundary Routing topology causes SNA packets to be dramatically reduced in size. Data compression reduces the cost associated with the WAN lines by compressing the size of SNA packets, which increases the rate at which the now-smaller SNA packets traverse the line. Data compression causes the WAN line to be used more efficiently, that is, the faster SNA traffic traverses the WAN line, the more bandwidth is available to route or forward more SNA packets.

For more information, see the Configuring Data Compression chapter.

Peer Data Exchange You can configure specific clients or *peers* on leaf networks in an IBM Boundary Routing topology to exchange data. For example, in the NetBIOS topology shown in Figure 296, imagine that LAN requesters B and C need to exchange data with each other, but they do not need to exchange data with LAN requesters A and D.

Figure 296 Peer Data Exchange In a NetBIOS Topology

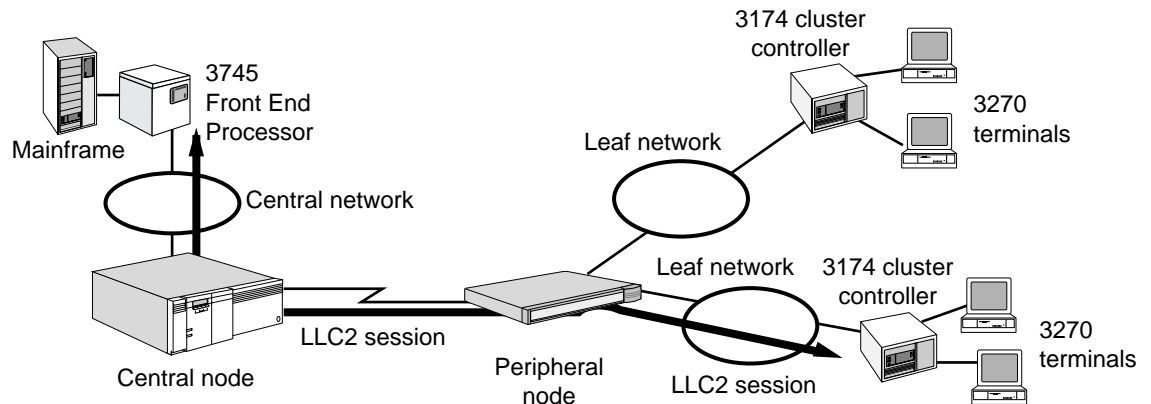
You can configure LAN requesters B and C to exchange data by setting the value of the `-LLC2 CONTROL` parameter to `Enable` on virtual ports V2 and V3 of the central node. The `-LLC2 CONTROL` parameter usually is enabled on LAN ports only, for example, on port 1 in the NetBIOS topology. By enabling this parameter on virtual ports V2 and V3 in this topology, you are essentially making these virtual WAN ports operate as LAN ports.

Increased Reliability

The features discussed in the following sections increase the reliability associated with WAN usage.

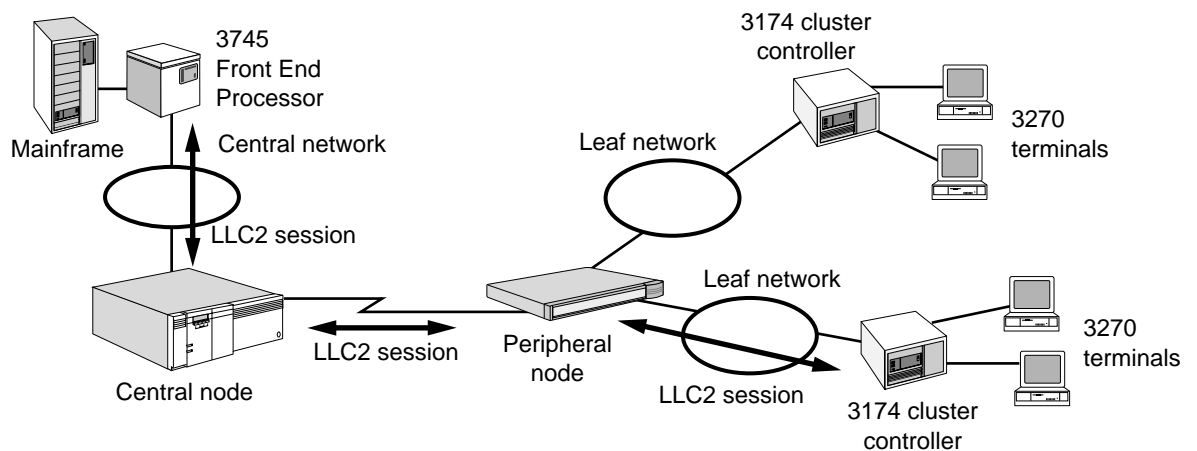
Local Termination In software versions 8.1 and earlier, LLC2 sessions initiated in an IBM Boundary Routing topology are considered *end-to-end*. End-to-end LLC2 sessions are those initiated at a terminal and run continuously from terminal, cluster controller, or LAN requester to the peripheral node, to the central node, and terminated at the front-end processor (FEP), mainframe, or LAN server. Figure 297 is an example of an end-to-end LLC2 session. The problem with this type of session is that many IBM applications running on an SNA or NetBIOS host are timing-sensitive. Delays or bottlenecks in the WAN can cause these applications to time out, which can cause users at terminals to lose data and to log on to the network again if the LLC2 session goes down.

Figure 297 End-to-End LLC2 Session



In software versions 8.2 and later, an LLC2 session initiated in the same IBM Boundary Routing topology is considered *logical end-to-end*. A logical end-to-end LLC2 session is one that is terminated on the local port of each 3Com bridge/router and boundary router and then reinitiated at the wide area port. Figure 298 shows an example of a logical end-to-end LLC2 session. In this figure, an LLC2 session is initiated at a terminal on one of the leaf networks. The session is locally terminated at the peripheral node. The peripheral node then initiates another session, which is terminated at the central node. The central node initiates another session, which terminates at the FEP.

Figure 298 Logical End-to-End LLC2 Session



The ability of the 3Com bridge/routers and boundary routers to terminate an LLC2 session on a local port and initiate another LLC2 session on a wide area port or vice versa is called *local termination*. In addition to breaking up a continuous LLC2 session into multiple sessions, local termination switches packets from an SNA or NetBIOS environment to a Boundary Routing environment and reduces the propagation of SNA and NetBIOS broadcast packets on the WAN.

Although a logical end-to-end LLC2 session is broken down into multiple sessions, these still provide a continuous logical link from terminal or workstation to host and vice versa. In fact, breaking a continuous end-to-end LLC2 session into multiple sessions eliminates delays or bottlenecks thereby making the session more reliable.

To activate local termination, set the value of the `-BCN CONTROL` parameter to `lbmTraffic`.

Automatic Prioritization of IBM Traffic Because of the interactive way in which clients and their SNA or NetBIOS hosts interoperate, IBM traffic has the following characteristics:

- It tends to be mission critical.
- It tends to be bursty.

To ensure the access of accurate information in the shortest amount of time possible, SNA and NetBIOS traffic that is sent through a port configured for Boundary Routing has been automatically prioritized as high and medium, respectively. No configuration is necessary.

Non-IBM protocols, such as IP, IPX, and AppleTalk traffic are also automatically prioritized as medium.

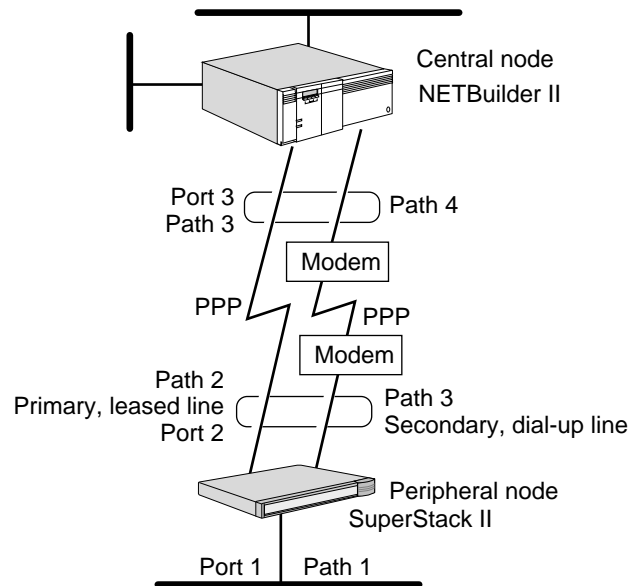
For example, if SNA, NetBIOS, and IP traffic must traverse wide area links that have been configured for Boundary Routing, automatic prioritization allows SNA traffic to travel across wide area links first, then NetBIOS or IP traffic, depending on which type of traffic is first in the queue.

Automatic prioritization of IBM traffic is a separate and distinct feature from the prioritization that the APPN class of service feature provides.

Continuous Operation

Boundary Routing software provides continuous operation with the dial-up backup line for disaster recovery or bandwidth-on-demand, with the assignment of network numbers, and with a mechanism for constructing resilient networks.

Dial-up Backup Line for Disaster Recovery or Bandwidth-on-Demand You can use dial-up paths to take advantage of disaster recovery or bandwidth-on-demand features in non-IBM and IBM Boundary Routing topologies. The dial-up paths must belong to the same port, must be connected to the same end-points, and must be running PPP as the data link protocol as shown in Figure 299. Although this figure shows Ethernet as the LAN media type, token ring can also be used.

Figure 299 Boundary Routing Backup Line

The secondary path to be used for disaster recovery or bandwidth-on-demand can be selected from the dynamic dial path pool.

Lines are monitored by *bandwidth management*, which applies static bandwidth, dynamic bandwidth, or a combination of these, to provide a port with the bandwidth it needs to meet current requirements. A line failure that drops the port bandwidth below a specified level causes bandwidth management to restore the specified bandwidth. If traffic conditions warrant additional bandwidth, the bandwidth-on-demand function also automatically increases the bandwidth accordingly. You can configure a line specifically for disaster recovery or as a general purpose (unrestricted) line that can be allocated for disaster recovery.

At the peripheral node, you need to assign two paths to one port, configure the path attributes for the lines, and enable disaster recovery and bandwidth-on-demand.

At the central node, you need to assign two paths to one port (or use the dynamic dial path pool), configure the path attributes for the lines, and enable disaster recovery or bandwidth-on-demand.

Enabling disaster recovery or bandwidth-on-demand allows bandwidth management to switch traffic to another path or allocate additional resources (disaster recovery) or allocate additional path resources if the traffic threshold on the path is exceeded (bandwidth-on-demand).

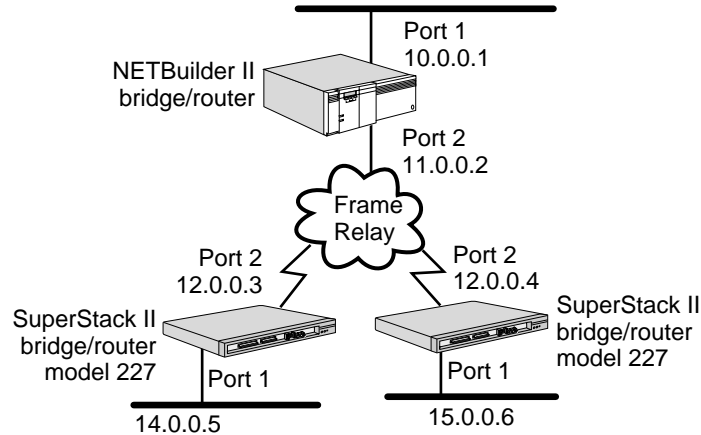
For information on configuring modems, see the *WAN Cabling and Connectivity Guide*. You can find this guide on the 3Com World Wide Web site by entering:

<http://www.3com.com/>

To configure backup dial-up lines for disaster recovery or bandwidth-on-demand on the central and peripheral node, see the Configuring Port Bandwidth Management chapter.

Assigning Network Numbers Assigning network numbers for routing protocols such as IP, IPX, and AppleTalk in a Boundary Routing topology differs from the same task in a non-Boundary Routing topology. For example, in the non-Boundary Routing topology using IP routing shown in Figure 300, an IP network number (IP address) is assigned to each LAN port and to each WAN port that is directly attached to the Frame Relay network.

Figure 300 Assigning Network Numbers in a Non-Boundary Routing Topology



In a Boundary Routing topology, assign network numbers to the following ports:

- The LAN port on the central node
- The virtual port on the central node for each remote site

The network number assigned to the virtual port is also used for the remote LAN. (A virtual cable connects the central node to the LAN connector on the peripheral node.) For information on administering IP addresses for peripheral nodes using either a Reverse Address Resolution Protocol (RARP) or BOOTP server, see "Configuring for PPP," "Configuring for Frame Relay," or "Configuring for X.25."

For example, in the Boundary Routing topology shown in Figure 301, network numbers for IP, IPX, and AppleTalk routing are assigned to the LAN port of the central node (port 1) and to each remote LAN through the use of virtual ports (virtual ports V2 and V3). The dashed lines in Figure 301 indicate the virtual connection between the central and peripheral nodes.

Figure 301 Assigning Network Numbers in a Boundary Routing Topology

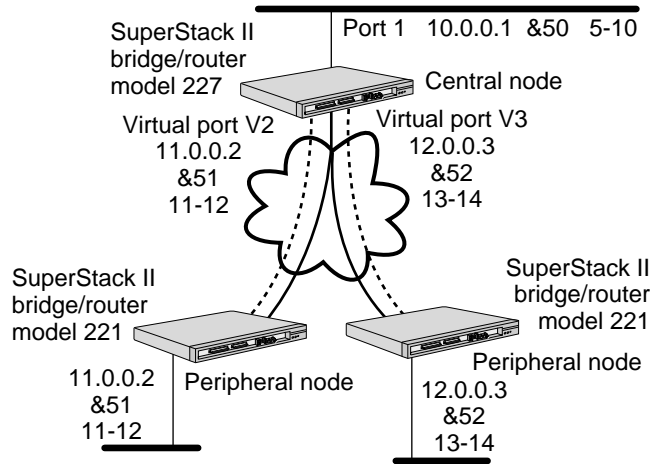


Table 76 lists the ports and virtual ports in Figure 301 and the network numbers assigned to them to help you understand how network numbers are assigned specifically for the IP, IPX, and AppleTalk protocols.

Table 76 IP, IPX, and AppleTalk Network Numbers For Central Node in Boundary Routing Topology

| Port Number (As Shown in Figure 301) | IP Network Number (Address) | IPX Network Number | AppleTalk Network Number (Range) |
|--------------------------------------|-----------------------------|--------------------|----------------------------------|
| Port 1 | 10.0.0.1 | &50 | 5 – 10 |
| Virtual port V2 | 11.0.0.2 | &51 | 11 – 12 |
| Virtual port V3 | 12.0.0.3 | &52 | 13 – 14 |

Dual PVCs for IBM Traffic

Dual PVCs can divide IBM and non-IBM traffic over a Frame Relay data link in a Boundary Routing environment. IBM traffic at a leaf node is directed to its own PVC and transmitted to a central site using Boundary Routing. Dual PVCs enhance response time and bandwidth available for IBM traffic and allow network managers to monitor the IBM data link separately.

Only virtual ports are used in the Boundary Routing Frame Relay environment. The DLCI numbers used for the PVCs are specified when the virtual ports are defined. You must define the DLCI pairs using the -BCN LclNonIbmDlci parameter. Specify the DLCI that will be used for IBM traffic at the leaf node using the -BCN RemNonIbmDlci parameter. Use the -PORT CONTROL parameter to enable the port, which transmits the PVC configuration information from the central node to the leaf node.

To separate IBM traffic from non-IBM traffic, the software uses SAP numbers to filter for IBM frames. The software assumes frames whose SAP numbers fall between 0 and 0xF0 and that are divisible by 4 are IBM frames. The exceptions to this assumption are VINES IP SAP frame number 0xBC, IPX SAP frame number 0xE0, and Sync Research special SAP frame number 0xFC.

Network Resiliency

Through hardware and software configuration, you can design a Boundary Routing topology that has a backup or redundant link between a central and peripheral node or a backup or redundant route to an alternate central node. See Figure 286 for an illustration of these two network resiliency configurations.

If your Boundary Routing topology has a redundant link between the central and peripheral nodes and the link fails, the central node will send and receive packets from the peripheral node through the redundant link. If your Boundary Routing topology has a redundant route to an alternate central node and the primary route or primary central node fails, the alternate central node will send and receive packets from the peripheral node through the redundant route.

The peripheral node allows only one active link between a central and peripheral node at a time. In a Boundary Routing topology with a redundant link, the primary link is considered the preferred link. In the topology with the redundant route to an alternate central node, the primary route is considered the preferred route.



When you turn the power on, the auto startup feature brings up one active path only. It does not bring up a second path for network resiliency. You need to configure the second path for network resiliency using the procedure in "Configuring Network Resiliency" earlier in this chapter.

The peripheral node software uses a set of precedence rules to determine which link or route should be treated as "preferred" when more than one link or route is available to be activated. These precedence rules are as follows:

- Leased-line connections take precedence over dial-up connections.
- Switched service (Frame Relay, X.25) takes precedence over PPP when they coexist. Within the switched services, Frame Relay takes precedence over X.25. PPP is likely to be used over a dial-up connection.

In operation, the precedence rules work as follows:

- When there is no currently active port, a port is allowed to come up.
- If the currently active port is a dial-up port and the new port is a leased-line port, then the leased-line port is allowed to come up and the dial-up port is deactivated.
- If the new port owner is a switched service, for example, Frame Relay, and the currently active port is point-to-point, then the Frame Relay port is allowed to come up and the PPP port is deactivated.
- In all other cases, the currently active port is left activated, and the new port is not allowed to come up.

In a Boundary Routing topology with a redundant route to an alternate central node, you must configure the alternate central node with the same address information as the primary central node. For more information, see "Primary and Alternate Central Node Configuration" later in this chapter.

If you plan to bridge or route AppleTalk, IP, or IPX, you must enable a central MAC address. You may also need to configure the alternate central node with the same bridging or routing attributes as the primary central node. For more information, see "Using the Central MAC Address" later in this chapter.

Network Resiliency Using a Redundant Link

You can configure a redundant link between a central and peripheral node in PPP and Frame Relay environments. You can also configure a redundant link in an IBM Boundary Routing topology that uses PPP or Frame Relay.

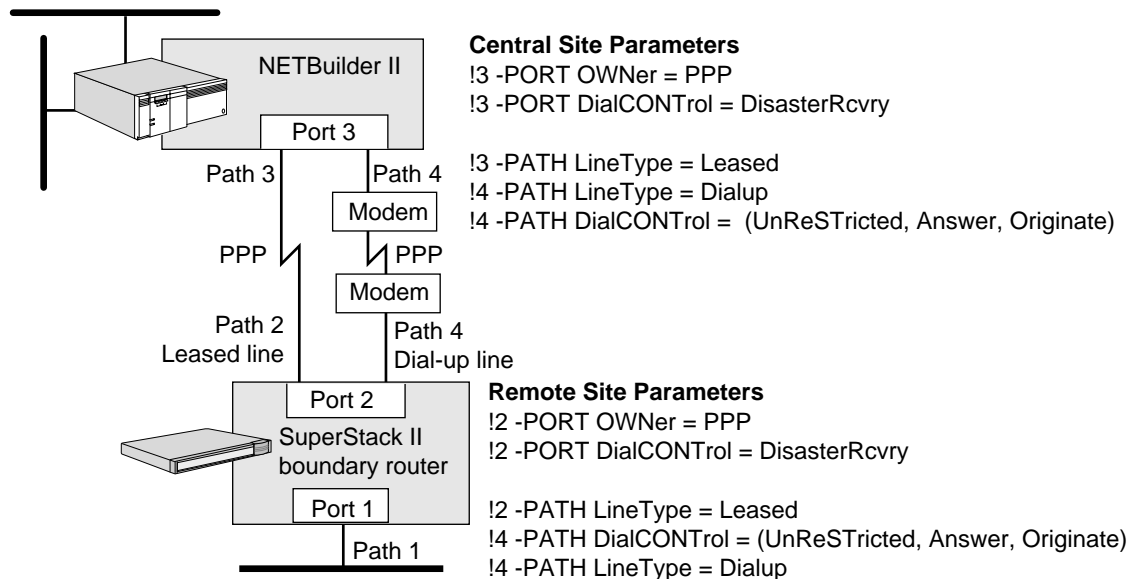
PPP Environment A primary link using a PPP leased line and a redundant link using a PPP dial-up line provides network resiliency in the event that the primary link fails at either the central or peripheral nodes.

To achieve this network resiliency in this configuration, two paths are mapped to a single logical port on both ends of the WAN link. For example, the devices can exchange data over a primary link with a secondary dial-up link for either disaster recovery or bandwidth-on-demand. Disaster recovery activates an additional dial-up line if the primary lines fails. Bandwidth-on-demand (through bandwidth management) activates additional resources in cases where the line experiences congestion. Achieving link redundancy in this way is supported only for PPP-based Boundary Routing environments (dial-up or leased lines), because a Frame Relay or X.25 port does not support multiple physical paths.

In the configuration discussed in the preceding paragraph, you can use either a DTE or ISDN line as the secondary dial-up link for bandwidth-on-demand and disaster recovery. For more information on ISDN, see the Configuring Wide Area Networking Using ISDN chapter.

The two links between the two nodes maintain the single connection to the central node, which is key to the Boundary Routing system architecture because logical data flows over a single WAN port. Figure 302 is an example of the PORT and PATH Service parameter values applicable to this configuration.

Figure 302 Network Resiliency PORT and PATH Parameters in a PPP Environment

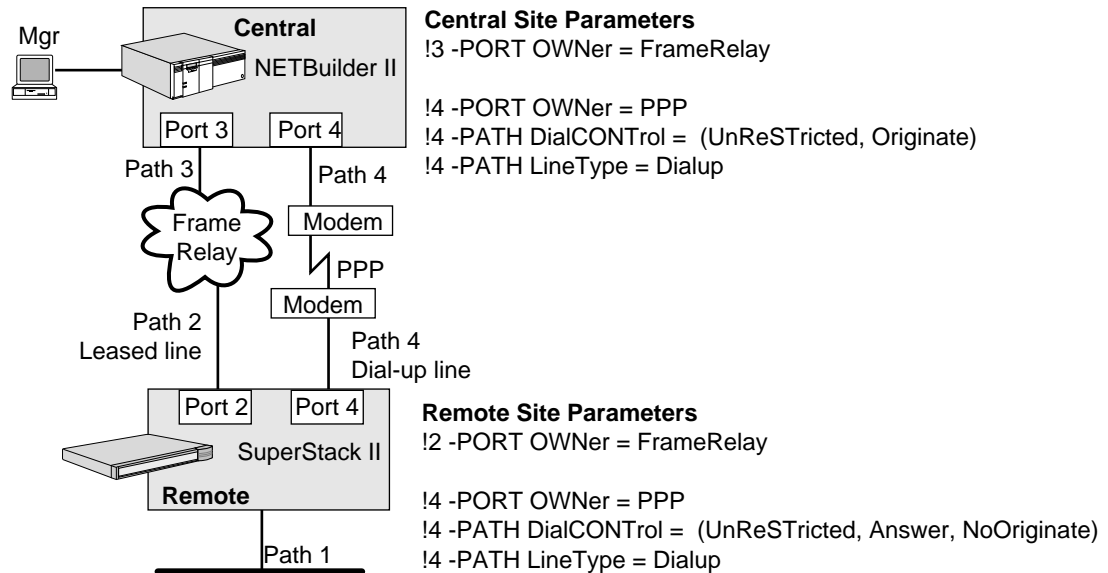


Frame Relay Environment A primary link using a Frame Relay leased line and a redundant link using a PPP dial-up line provides network resiliency if the primary link fails at either the central or peripheral nodes.

In the configuration discussed in the preceding paragraph, a data terminal equipment (DTE) or Integrated Services Digital Network (46) line can be used as the backup dial-up line for network resiliency.

Figure 303 is an example of the PORT and PATH Service parameters that support Frame Relay as the primary link and PPP as the redundant (dial-up) link.

Figure 303 Network Resiliency PORT and PATH Parameters in a Frame Relay Environment



This configuration differs from the configuration discussed in “PPP Environment” earlier in this chapter because instead of allowing multiple paths per port as PPP does, Frame Relay requires that a single path and single port mapping must be maintained. Two logical port destinations are possible on the central site router instead of just one as in the PPP configuration.

In this configuration, network resiliency is achieved with additional user or SNMP intervention. Each interface to which the central node is connected must be configured with the identical network address information. Because duplicate network addresses are not allowed on the same NETBuilder II bridge/router or model 227, 327, 427, or 527 SuperStack II bridge/router, macros can be predefined to delete the network address on the Frame Relay port and add that same address to the PPP port as required. When the loss of connection to the remote site is detected, the macro executes to properly address the backup port. A similar macro may be created to reverse this process to change back to the primary port when it recovers.

Macros can be executed manually, or a central site management station can automate the process by monitoring the status of the remote site. When the user or the management station detects that the remote site is no longer reachable, the user or the management station may run a script file that Telnets to the central node and executes the macro. This operation will cause a session disruption.

The central site macro that activates the redundant link must follow these steps:

- Disable the primary port.
- Delete addresses from the primary port.
- Add addresses to the backup port.
- Enable the backup port.

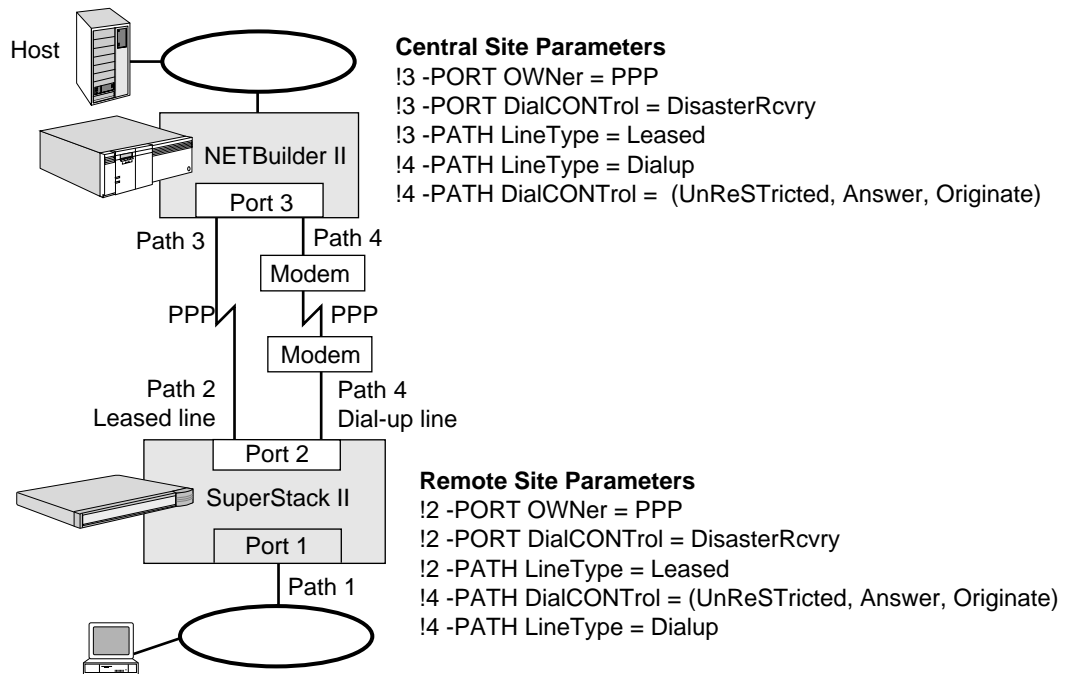
The central site macro that reactivates the primary link must follow these steps:

- Disable the backup port.
- Delete addresses from the backup port.
- Add addresses to the primary port.
- Enable the primary port.

IBM Environment You can configure a redundant link in an IBM Boundary Routing topology that uses PPP or Frame Relay.

Figure 304 shows an IBM Boundary Routing topology that has a primary link using a PPP leased line and a redundant link using a PPP dial-up line.

Figure 304 Network Resiliency PORT and PATH Parameters Using PPP



To achieve network resiliency in this configuration, two paths are mapped to a single logical port on both ends of the WAN link. For example, the devices can exchange data over a primary link with a secondary dial-up link for both disaster recovery and bandwidth-on-demand. Disaster recovery activates the secondary, dial-up line if the primary lines fails. Bandwidth-on-demand activates the secondary line in cases where the primary line experiences congestion. Achieving link redundancy in this way is supported only for PPP-based Boundary Routing environments (dial-up or leased lines), because a Frame Relay or X.25 port does not support multiple physical paths.

In the IBM Boundary Routing topology with a primary PPP leased line and the secondary PPP dial-up line, you can use a DTE or ISDN line as the secondary dial-up link for disaster recovery and bandwidth-on-demand.

A problem experienced when performing Boundary Routing in a connection-oriented environment such as SNA and NetBIOS is that when a currently active port deactivates and a new port activates, the session between the central and peripheral nodes is disrupted. In the IBM Boundary Routing topology

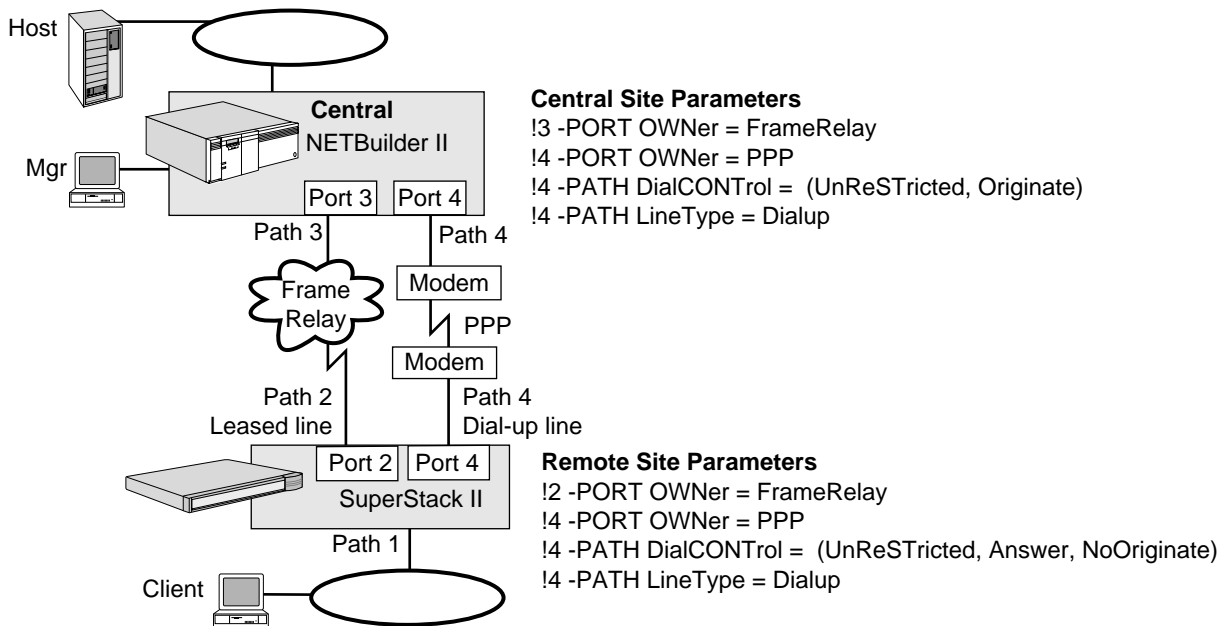
with a primary PPP leased line and the secondary PPP dial-up line, you will not experience disruption for two reasons:

- This topology requires that two paths are mapped to one port. The local termination feature, which is activated when Boundary Routing over IBM is enabled, isolates the mainframe or LAN server and terminal or LAN requester from disruptions between the central and peripheral nodes.
- Increasing the retry counter on the Boundary Routing port of the central node decreases the possibility that the circuit will be brought down while the current active port deactivates and the new port activates.

Having two links between the two nodes maintains the single connection to the central node that is key to the Boundary Routing system architecture, because logical data flows over a single WAN port.

Figure 305 shows an IBM Boundary Routing topology that has a primary link using a Frame Relay leased line and a redundant link using a PPP dial-up line. You can use a DTE or ISDN line as the backup dial-up line for network resiliency.

Figure 305 Network Resiliency PORT and PATH Parameters Using Frame Relay



As with the IBM Boundary Routing topology using PPP discussed earlier in this section, the LLC2 session between client and host is disrupted if the primary link fails and the redundant link activates in IBM Boundary Routing topology using Frame Relay.

In the IBM Boundary Routing topology using Frame Relay, network resiliency is achieved with additional user or SNMP intervention. Each interface to which the central node is connected must be configured with the identical network address information. Because duplicate network addresses are not allowed on the same NETBuilder II bridge/router or model 227, 327, 427, or 527 SuperStack II bridge/router macros can be predefined to delete the network address on the Frame Relay port and add that same address to the PPP port as required. When the loss of connection to the remote site is detected, the macro executes to properly address the backup port. A similar macro may be created to reverse this process in order to change back to the primary port when it recovers.

Macros can be executed manually, or a central site management station can automate the process by monitoring the status of the remote site. When the user or the management station detects that the remote site is no longer reachable, the user or the management station may run a script file that Telnets to the central node and executes the macro. This operation will cause a session disruption.

The central site macro that activates the redundant link must follow these steps:

- Disable the primary port.
- Delete addresses from the primary port.
- Add addresses to the backup port.
- Enable the backup port.

The central site macro that reactivates the primary link must follow these steps:

- Disable the backup port.
- Delete addresses from the backup port.
- Add addresses to the primary port.
- Enable the primary port.

Network Resiliency Using a Redundant Route to an Alternate Central Node

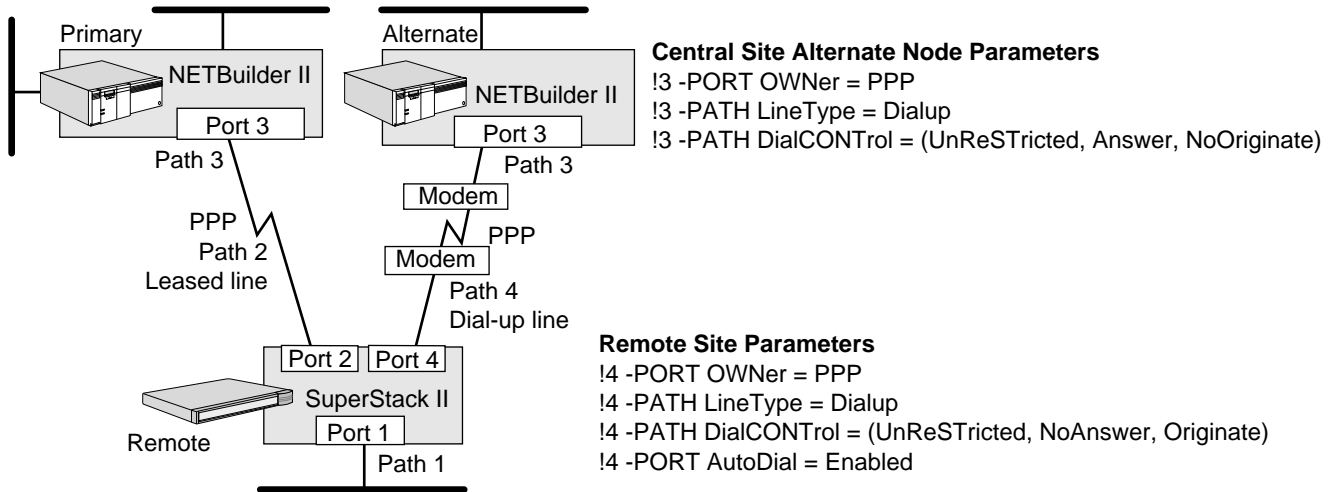
You can configure a redundant route to an alternate central node in PPP, Frame Relay, and X.25 environments. You can also configure a redundant route to an alternate central node in an IBM Boundary Routing that uses PPP, Frame Relay, or X.25.

The precedence rules discussed in “Network Resiliency” earlier in this chapter apply at the port level, so when two central nodes are used, a single path and single port mapping must be maintained on both central and peripheral nodes, regardless of the media combination. When using two central nodes, network resiliency does not work properly with multiple paths assigned to a single port.

PPP Environment A primary route using a PPP leased line and a redundant route using a PPP dial-up line provides network resiliency if the primary route fails at either the primary central or peripheral nodes or if the primary central node fails. You can use a DTE or ISDN line as the backup dial-up line for network resiliency. For more information on ISDN, see the Configuring Wide Area Networking Using ISDN chapter.

Figure 306 shows the PORT and PATH Service parameters required by this configuration.

Figure 306 Network Resiliency Parameters for Two Central Site Nodes in a PPP

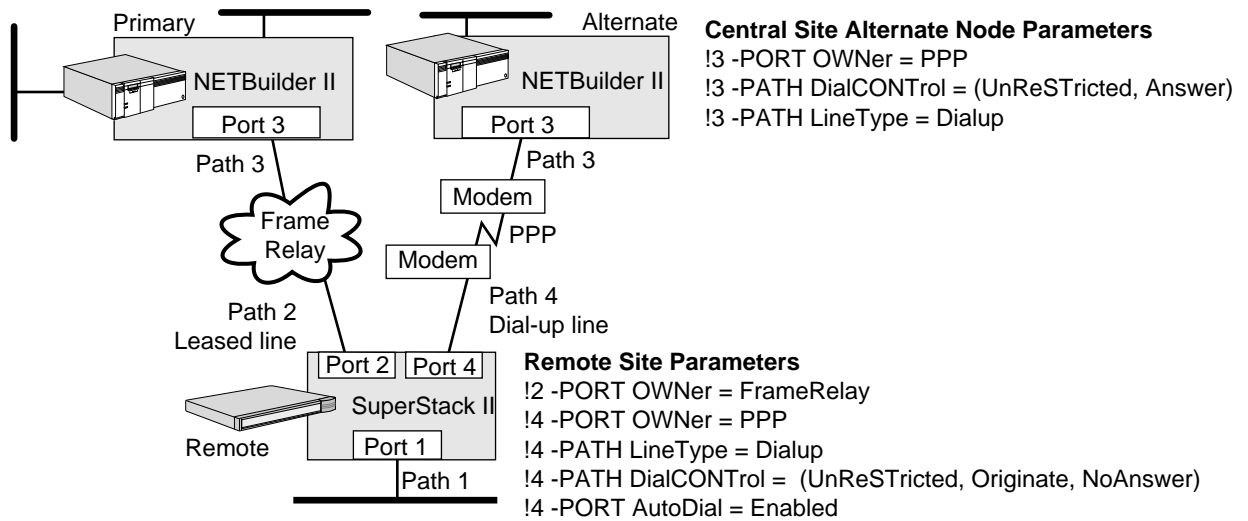


Environment

The precedence rules ensure that a leased line is always active when available. In the case of a failure, the dial-up line is activated and the peripheral node attempts to connect to the alternate central site. When the leased line recovers, it again takes precedence over the dial-up line. The peripheral node automatically hangs up the connection and changes back to the leased line.

The precedence rules applied at the remote site determine which connection to keep active. In this case, the connection attempts are fully automated and generated from the peripheral node instead of from the central site.

Frame Relay Environment A primary route using a Frame Relay leased line and a redundant route using a PPP dial-up line provides network resiliency if the primary route fails at either the primary central or peripheral nodes or if the primary central node fails. You can use a DTE or ISDN line as the backup dial-up line for network resiliency. For more information on ISDN, the *Configuring Wide Area Networking Using ISDN* chapter. Figure 307 shows the PORT and PATH Service parameters required by this configuration.

Figure 307 Network Resiliency Parameters for Two Central Site Nodes in a Frame Relay Environment

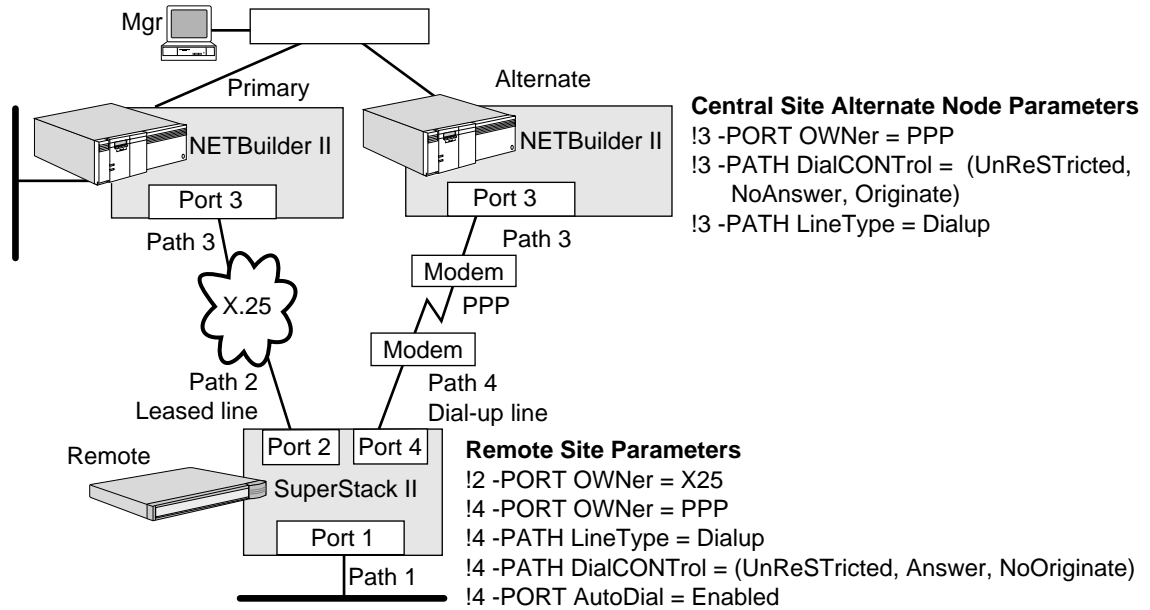
The precedence rules ensure that the Frame Relay line is always active when available. In the case of a failure, the PPP line is activated and the peripheral node dials the alternate central site node. When the Frame Relay line recovers, it again takes precedence over the PPP line and the peripheral node automatically hangs up the connection and changes back to the Frame Relay link.

In case of a primary route or primary central node failure, the Local Management Interface (LMI) Protocol will no longer report the primary central node data link connection identifier (DLCI) to the peripheral node. This triggers a path-down state at the peripheral node at which time the PPP dial backup attempts to dial the alternate central node. The precedence rules cause this operation to be controlled at the peripheral node.

X.25 Environment A primary route using an X.25 leased line and a redundant route using a PPP dial-up line provides network resiliency in the event that the primary route fails at either the primary central or peripheral nodes or if the primary central node fails. You can use a DTE or ISDN line as the backup dial-up line for network resiliency. For more information on ISDN, see the Configuring Wide Area Networking Using ISDN chapter.

Figure 308 shows the PORT and PATH Service parameters required by this configuration.

Figure 308 Network Resiliency Parameters for Two Central Site Nodes in an X.25 Environment



Network resiliency in an X.25 environment requires manual intervention unless triggered by a management station. X.25 supports switched virtual circuits (SVCs), and circuits are established and torn down as required by traffic flow through the central node. The loss of an X.25 virtual circuit does not cause a path and port down state to occur at the central node. If the peripheral node loses its link to the X.25 network, the virtual port of the central node remains active. If the peripheral node is allowed to automatically dial an alternate central node, the connection is accepted, but the primary central node continues to assert its network layer information for the virtual port onto surrounding networks through routing updates. When the alternate central node accepts the incoming call, it also begins to assert the same routing information onto the network. Two routes are advertised to get to the remote LAN, but only one route is valid.

To avoid this situation, the virtual port on the primary central node must be disabled before the port on the alternate central node is allowed to accept the call. You can disable the virtual port using an SNMP management station. When the management station detects that it can no longer reach the peripheral node, it can execute a script that Telnets to the primary central node and disables the virtual port, then Telnets to the alternate central node and executes the connection attempt to the peripheral node. Special configuration parameters are required to ensure that a call cannot be established until this occurs.

IBM Environment You can configure a redundant route to an alternate central node in an IBM Boundary Routing topology that uses PPP, Frame Relay, or X.25. Figure 309, Figure 310, and Figure 311 show IBM Boundary Routing topologies that have primary routes using PPP, Frame Relay, and X.25 leased lines, respectively, and redundant routes using PPP dial-up lines.

Figure 309 Network Resiliency Parameters for Two Central Nodes in an IBM Topology Using PPP

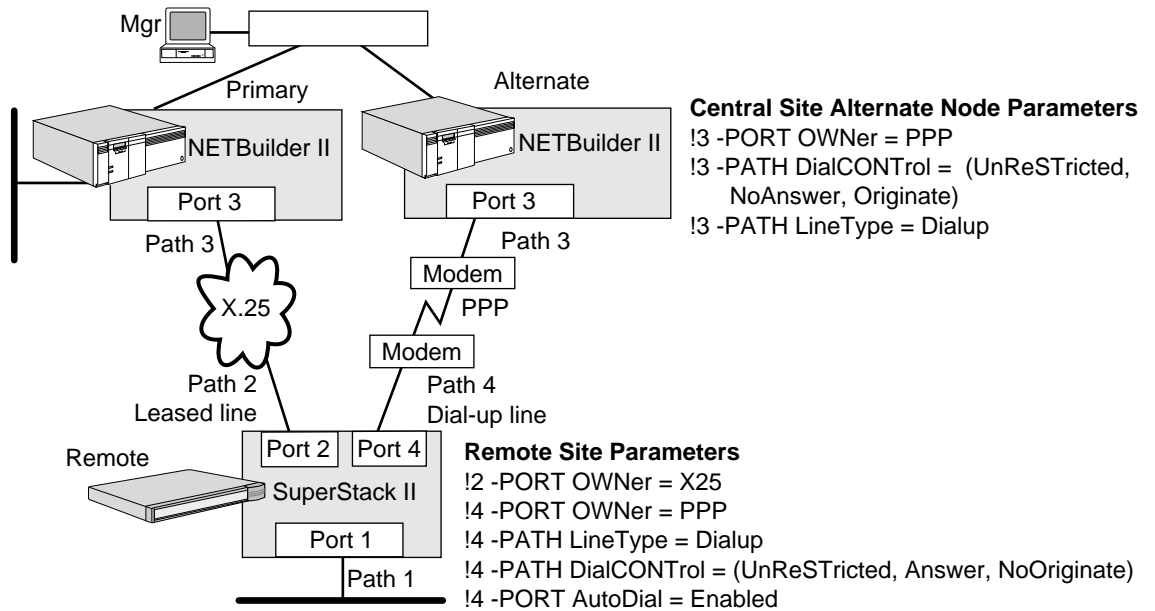


Figure 310 Network Resiliency Parameters for Two Central Nodes in an IBM Topology Using Frame Relay

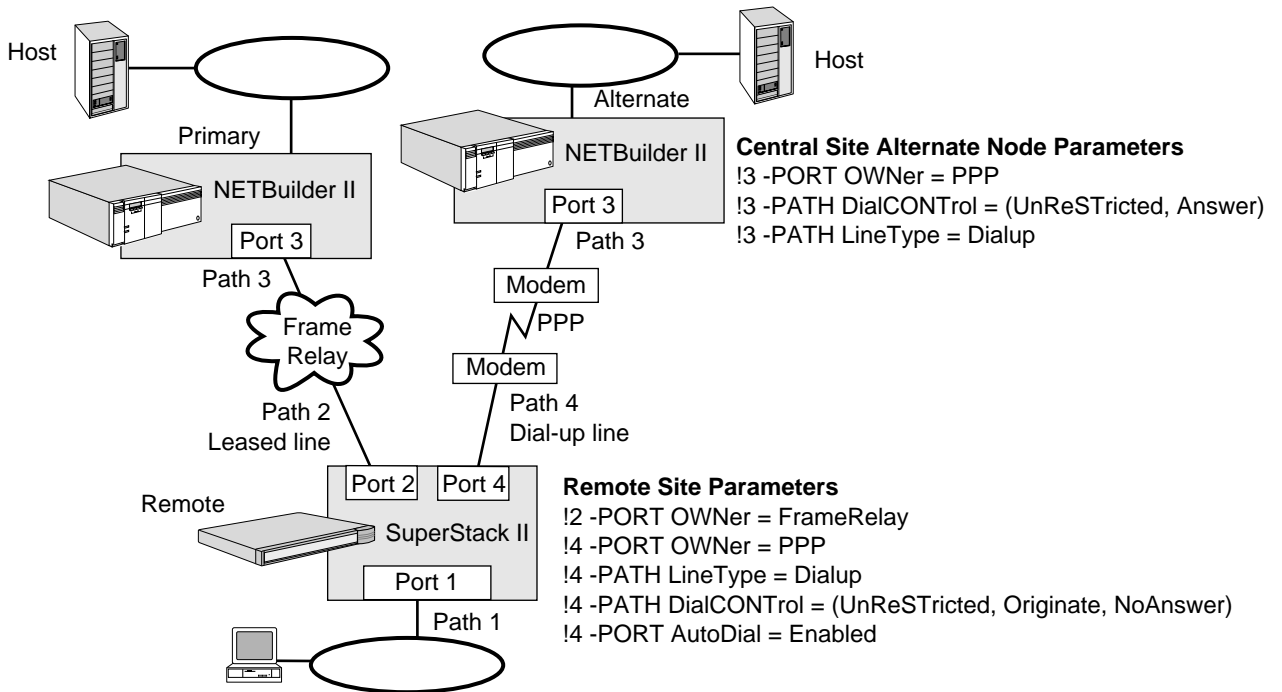
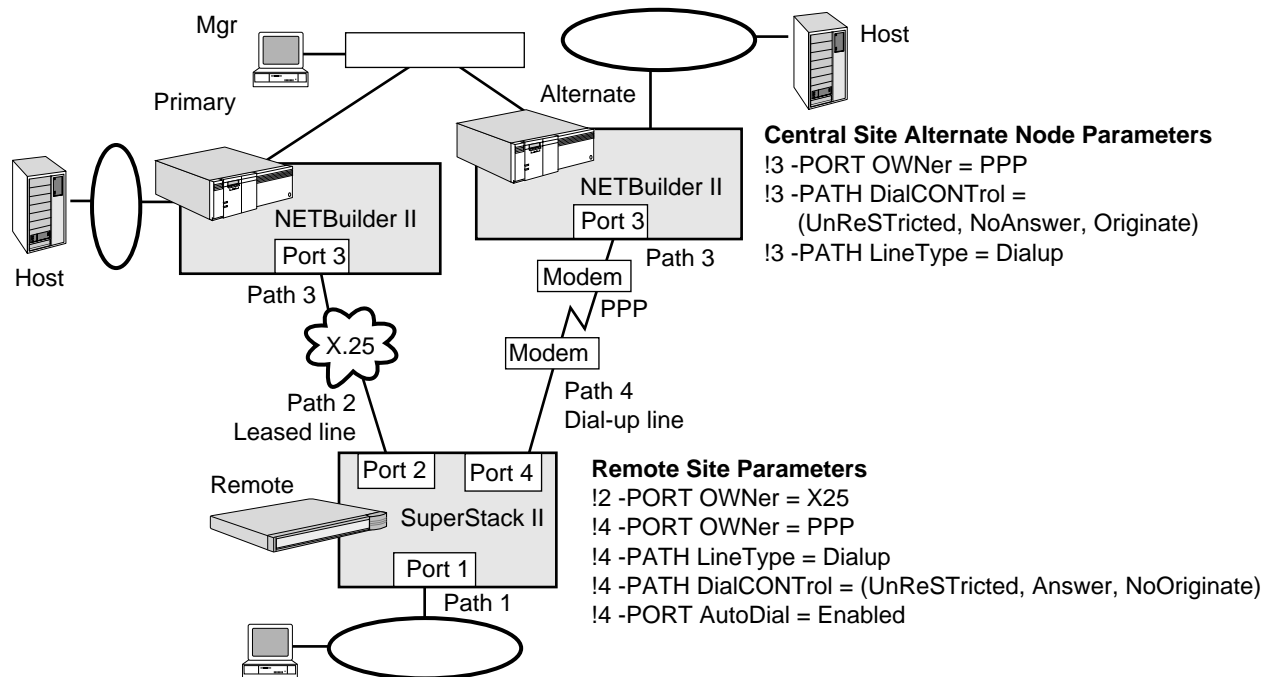


Figure 311 Network Resiliency PORT and PATH Parameters for Two Central Nodes in an IBM Topology Using X.25

In the topologies shown in each of these figures, you can use a DTE or ISDN line as the backup dial-up line used as the redundant route. For more information on ISDN, see the Configuring Wide Area Networking Using ISDN chapter.

In the topologies that have primary routes using PPP and Frame Relay leased lines, the precedence rules ensure that the PPP and Frame Relay leased lines are always active when available. In the case of a failure, the PPP dial-up line is activated and the peripheral node dials the alternate central site node. When the PPP or Frame Relay leased line recovers, it again takes precedence over the PPP dial-up line and the peripheral node automatically hangs up the connection and changes back to the PPP or Frame Relay leased line.

In the topology that has a primary route using PPP, the precedence rules applied at the remote site determine which connection to keep active. In this case, the connection attempts are fully automated and generated from the peripheral node instead of from the central site.

In the topology that has a primary route using Frame Relay, if a primary route or the primary central node fail, the LMI Protocol will no longer report the primary central node DLCI to the peripheral node. This triggers a path-down state at the peripheral node at which time the PPP dial backup attempts to dial the alternate central node. The precedence rules cause this operation to be controlled at the peripheral node.

In the topology that has a primary route using X.25, if the primary route or the primary central node fail, manual intervention is required unless you have previously generated scripts on your SNMP management station that automates certain tasks.

X.25 supports switched virtual circuits (SVCs), and circuits are established and torn down as required by traffic flow through the central node. The loss of an X.25

virtual circuit at the peripheral node does not cause a path and port down state to occur at the central node. If the peripheral node loses its link to the X.25 network, the central node virtual port remains active. If the peripheral node is allowed to automatically dial an alternate central node, the connection is accepted, but the primary central node continues to assert its network layer information for the virtual port onto surrounding networks through routing updates. When the alternate central node accepts the incoming call, it also begins to assert the same routing information onto the network. Two routes are advertised to get to the remote LAN, but only one route is valid.

To avoid this situation, the virtual port on the primary central node must be disabled before the port on the alternate central node is allowed to accept the call. This can be done through an SNMP management station. When the management station detects that it can no longer reach the peripheral node, it can execute a script that Telnets to the primary central node and disables the virtual port, then Telnets to the alternate central node and executes the connection attempt to the peripheral node. Special configuration parameters are required to ensure that a call cannot be established until this occurs.

Because the configuration of duplicate MAC addresses is not used in an IBM Boundary Routing environment, if you use one of the topologies discussed in the preceding paragraphs, you will experience a disruption if one port deactivates and another activates. When the session has been disrupted, you will need to log in again and reinitiate a session.

In each of the topologies, imagine that the client has initiated an LLC2 session with the host on the primary central network. Since the primary line is up, the session takes place over this line. If the primary line goes down during the session between client and host, from the user's perspective, the session abruptly terminates or is disconnected. Eventually, the secondary line comes up. If you attempt to log in and reinitiate a session with the host before the secondary line comes up, you will be unsuccessful; if the attempt is made after the secondary line comes up, the attempt will be successful. When the primary line has been repaired and comes up again, the disruption will occur again.

Primary and Alternate Central Node Configuration End stations on the remote LAN use the logical address of the primary central node WAN port as the next hop when routing data. To provide a transition to the alternate central node, WAN ports on both routers connected to a remote site in non-IBM and IBM Boundary Routing topologies must share the same address information. Because the Boundary Routing system architecture does not allow both connections to be active at the same time, it is possible to configure identical logical addresses (IP addresses, IPX network number, and so forth) on both routers. The address duplication does not interfere with network operation as long as the connections are not simultaneously active.

Some non-IBM protocols, such as IPX, present more of a challenge because they adopt the underlying MAC addresses for use as a logical host address. Duplicated network numbers are not sufficient in this situation; you must also configure the central node to use the same MAC address on those WAN interfaces. For information on configuring both primary and alternate central nodes to use the same MAC address on each WAN port, see "Using the Central MAC Address" next.

Using the Central MAC Address

If you configure a redundant route to an alternate central node, you may need to configure both primary and alternate central nodes to use a central MAC address, which is a special, internally saved MAC address. This MAC address allows certain protocols to switch to the alternate central node without losing sessions between a client on the leaf network and a host on the central site network.



If you are configuring disaster recovery over Frame Relay on the peripheral node of a Boundary Routing environment, do not configure the central MAC address.

Use the central MAC address with the following protocols:

- IPX
- AppleTalk
- IP
- Bridging

Setting the `-BCN CONTROL` parameter to `CentralMac` on the Boundary Routing ports of both the primary and alternate central nodes causes both nodes to use the same MAC address. The transition to an alternate central node, if necessary, is completely transparent to the end stations.

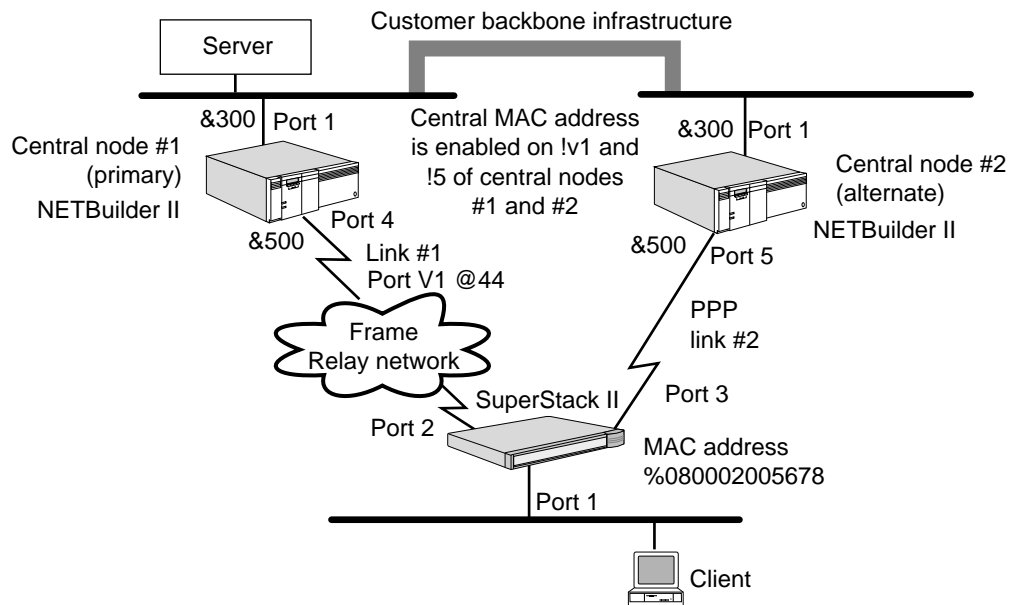
The central MAC address is not used in an IBM Boundary Routing environment.

IPX Routing Example When routing IPX in a Boundary Routing environment as shown in Figure 312, you need to configure the alternate central node #2 with the same routing attributes and network addresses as central node #1. You must also enable IPX routing on both central nodes. If you enable the use of the central MAC address on both central nodes, the switch from the primary to alternate central node is transparent to the user.

For example, if a user on the leaf network has a session established with a server on the central site network when a failure occurs (link #1 or central node #1 fails), the alternate central node brings up link #2, and the session between the client and the server is not disrupted. The client continues the session with the server although the route is established on link #2 and through the alternate central node.

The use of the central MAC address provides a transparent switch to the alternate central node because NetWare clients cache next-hop router information, including the router's MAC address and network number. In addition, clients do not listen to RIP updates to validate router addresses or detect routers that have gone down. By configuring the same network addresses and enabling the central MAC address on both central nodes, if link #1 or central node #1 fails when a session is established between a client on the leaf network and a server on the central site network, the session is not interrupted when the alternate central node becomes active. The client continues to use the MAC address and network address that is stored in its cache, although the client is accessing the server through link #2 and alternate central node #2.

If you do not enable the use of the central MAC address on both of the central nodes, the session is disrupted and manual reconnection and login to the server is required.

Figure 312 Network Resiliency with IPX Routing in a Boundary Routing Environment

AppleTalk Routing Example When routing AppleTalk in a Boundary Routing environment, you need to configure the alternate central node #2 with the same network range, default zone, seed information, and zone list as central node #1. If you enable the use of the central MAC address on both central nodes, the switch from the primary to the alternate central node is transparent to the user.

The use of the central MAC address provides a transparent switch to the alternate central node. If link #1 or central node #1 fails when a session is established between a client on the leaf network and a server on the central site network, the session is not interrupted when the alternate central node becomes active because the same routing and addressing attributes are used on central node #2.

If you do not enable the use of the central MAC address on both of the central nodes, the session is disrupted. Because AppleTalk clients identify their next-hop router by listening to Routing Table Maintenance Protocol (RTMP) packets, their routing tables are eventually updated with a route to central node #2; however, you need to reconnect and log on to the server.

IP Routing Example When routing IP in a Boundary Routing environment, you need to configure the alternate central node #2 with the same network addresses and routing attributes as central node #1. In addition, you must enable IP routing on both central nodes. If you enable the use of the central MAC address on both central nodes, the switch from the primary to the alternate central node is transparent to the user.

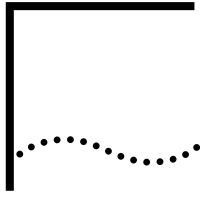
The use of the central MAC address provides a transparent switch to the alternate central node when clients on the leaf network have a single default gateway configured. These clients use ARP to obtain the gateway address and cache its MAC address. The clients may or may not use unsolicited ARP responses to update their caches. However, by configuring the same network addresses and enabling the central MAC address on both central nodes, if link #1 or central node #1 fails when a session is established between a client on the leaf network and a server on

the central site network, the session is not interrupted when the alternate central node becomes active. The client continues to use the MAC address and network address that is stored in its cache, although the client is accessing the server through link #2 and alternate central node #2.

If you do not enable the use of the central MAC address on both of the central nodes when the clients have a single default gateway configured, the session is disrupted. You need to reconnect and log on to the server.

Transparent Bridging Example When bridging in a Boundary Routing environment, you need to enable transparent bridging on both central nodes. If you enable the use of the central MAC address on both central nodes, the switch from the primary to the alternate central node is transparent to the user.

If link #1 or central node #1 fails when a session is established between a client on the leaf network and a server on the central site network, the session is not interrupted when the alternate central node becomes active. Packets are automatically forwarded on link #2, assuming that the Spanning Tree Protocol is active so that a previously blocked path to the destination is unblocked, and that activation of link #2 occurs before the application's session times out.



CONFIGURING AUTOSTARTUP

This chapter describes the required tools and prerequisites, how to configure the autostartup feature, and how autostartup works.

To use the Autostartup feature in Enterprise OS software release version 11.2, you must upgrade the remote node, the central site, and the 3Com BootP server (if being use). (Autostartup works with a non-3Com BootP server if the remote node is identified by MAC address.)

You can use the autostartup feature in conjunction with the capabilities of the ASCII boot files to streamline configuring devices on your network. For information on configuring using ASCII text files, see the Configuring with ASCII Files chapter.

- For information on setting up and the required tools, see “Prerequisites and Tools” .
- For information on setting up the BootP and TFTP servers, see “Configuring the Central Site Network Management Station” .
- For information on autostarting the central site device, see “Autostarting the Central Site Node” .
- For information on how autostartup works, see “How Autostartup Works” .

Prerequisites and Tools

The autostartup feature enables 3Com devices to boot and become operational with no software configuration on the device itself.



Before configuring autostartup, 3Com strongly recommends that you read “How Autostartup Works” later in this chapter.

For the autostartup feature to work, you need to set up a network management station at a central site location. On the central site network you must also set up a BootP and a Trivial File Transfer Protocol (TFTP) server.

You can set up your network to automatically startup the central site bridge/router (with the exception of a NETBuilder II bridge/router with a DPE or a PathBuilder switch). The central site bridge/router then in turn assists in automatically starting whatever peripheral nodes may be contained in the configuration on the network management station.

Table 77 lists the Enterprise OS devices that can be used as a central node. All Enterprise OS devices can be used as a remote node, with the exception of a NETBuilder II bridge/router or a PathBuilder switch.

Table 77 3Com Central Node Devices

| Central Node Type | Models and Software Packages |
|-----------------------------|--|
| NETBuilder II | All models with AC, DW, or DE packages |
| SuperStack II NETBuilder SI | Model 437, 447, 457, or 467 with CF or CE packages |
| PathBuilder | All S5xx tunnel switches |

Preparation For the autostartup feature to work, the central node requires certain software tools and some software configuration.

Tools

You need to configure a BootP server and a TFTP server on the central site network. Table 78 lists the software tools that offer the servers, indicates if they are mandatory or optional, and provides a short explanation of each tool.

Table 78 Software Tools to Configure Autostartup Phase 2

| Software Tool | Usage | Explanation |
|---|------------|--|
| 3Com's Upgrade Utilities (version 11.2 or higher) | Optional* | Compatible with the Solaris 2.x, and HP-UX 10.x, IBM AIX 4.x, and Wintel PC environments. These utilities include a BootP server required for 3Com proprietary BootP reply. |
| BootP server | Mandatory* | On UNIX systems only, executing the <code>bcmsetup -BootP</code> utility sets up the 3Com BootP server on the NMS where the Upgrade Utilities are running. |
| TFTP server (3CServer) | Mandatory* | The TFTP server (3CServer) application runs on any platform. It acts as a repository of firmware, software, and configuration files. A TFTP server is a part of the UNIX system software provided with the Upgrade Utilities.

Installing the Upgrade Utilities automatically sets up the 3Com TFTP server on Windows systems. |

* Use either 3Com's or another vendor's version of the BootP and TFTP servers.

For information on configuring the BootP and TFTP servers, see "Configuring the Central Site Network Management Station" next.

Prerequisites

Before beginning this procedure, complete the following tasks on the central node if autostartup will be through a central node. If the bridge/router being autostarted is connected to the central site network management station through a local area network, these prerequisite steps are not necessary:

- Log on with Network Manager privilege.
- Set up ports and paths according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter. Note the port numbers and associated IP addresses that will be used for the autostartup phase 2 connection.
- Configure a WAN port over which the peripheral nodes will communicate.
- Examine your network and determine (or assign) the IP address and/or the MAC address for each bridge/router that requires autostartup support.
- Determine on which server the BootP server will reside.
- Determine on which server the TFTP server will reside.
- For Frame Relay configurations, determine the data link connection identifier (DLCI) for each SuperStack II remote router that requires autostartup phase 2 support. The DLCI is assigned to the Frame Relay interface by the public data network (PDN) service vendor.
- If using a peripheral node with a token ring interface, configure ring and bridge numbers, which will be downloaded from the central node to the peripheral node, using the -SR RingNumber and -SR BridgeNumber parameters. For more information on these parameters, see *Reference for Enterprise OS Software*.



The DLCI is normally learned automatically by the interface. However, for the autostartup feature to work properly, you may need to include this value in the BootPtab file entry if you are using the 3Com Upgrade Utilities.

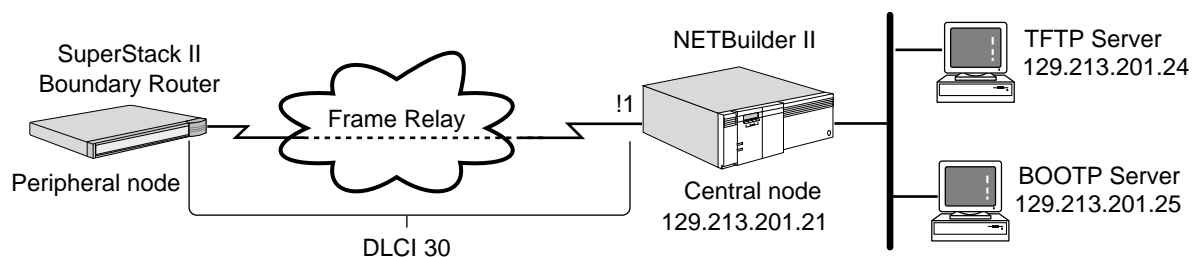
Configuring the Central Site Network Management Station

Figure 313 and Figure 314, show sample topologies in which autostartup is configured for Frame Relay and PPP.

The following items are the same in all the sample topologies:

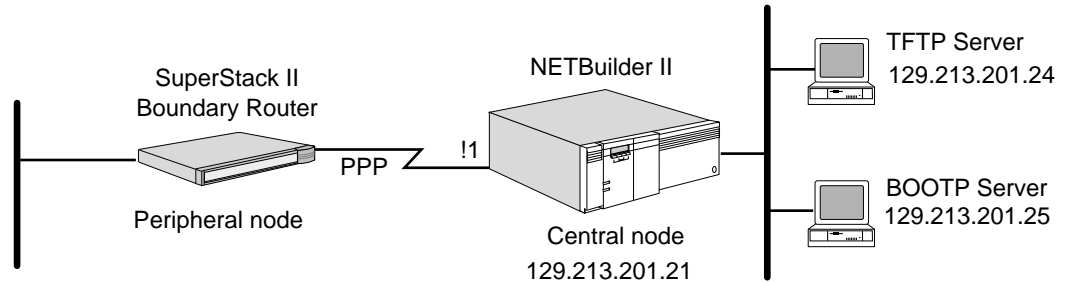
- A BootP server and a TFTP server are set up on Sun, HP, AIX, or Wintel PC systems. The BootP server has the IP address 129.213.201.25; the TFTP server has the IP address 129.213.201.24.
- The central node is functioning as a bridge/router with an IP address of 129.213.201.21. The User Datagram Protocol (UDP) Broadcast Helper feature has been enabled on it.

Figure 313 Configuring Autostartup for Frame Relay



The Frame Relay connection between the central and peripheral nodes in Figure 313 has a DLCI of 30 assigned to it by the Frame Relay service vendor.

Figure 314 Configuring Autostartup for PPP



Procedure To configure the central node and the BootP and TFTP servers on the central site network, follow these steps:

- 1 Configure the UDP Broadcast Helper on the central node.
 - a Enable UDP Broadcast Helper by entering:


```
SETDefault -UDPHELP CONTROL = Enable
```
 - b Add BootP server UDP port 67 to the active port list using:


```
ADD -UDPHELP ActivePorts {<UDP port> | <name>}
```

BPSERVER is the name reserved for port 67.

For example, to add UDP port 67 to the active port list, enter:

```
ADD -UDPHELP ActivePorts 67
```

OR

```
ADD -UDPHELP ActivePorts BPSERVER
```
 - c Add the IP address of the BootP server into the forward address list using:


```
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
```

For example, in the sample topologies shown, add the IP address of the BootP server (129.213.201.25) to the forward address list by entering:

```
ADD -UDPHELP ForwardAddress 67 129.213.201.25
```

For more information on the UDP Broadcast Helper, see the Configuring UDP Broadcast Helper chapter.
- 2 Install either 3Com's or another vendor's version of the BootP server on your Sun, HP, AIX, or Wintel PC system. If you plan to install another vendor's version of the BootP server, skip this step and see the documentation that accompanies that product for information.
 - a Edit the /etc/bootptab file in the BootP server file directory.

The /etc/bootptab file contains configuration parameters that must be set up before a bridge/router can execute phase 2 of the autostartup process.

The bootptab file has a format similar to that of the termcap file in which two-character, case-sensitive tag symbols are used to represent parameters. The parameter declarations are separated by colons (:). The general format is as follows:

```
hostname:tg=value.....:tg=value.....:tg=value...
```

where:

`hostname` is the actual name of a BootP client (the peripheral node).
`tg` is a two-character tag symbol. Most tags must be followed by an equal sign (=) and a value.

You can access a complete description of the `bootptab` file and its construction using the online manual page facility that comes with the utilities package.

Read the contents of the `bootptab` file. At the end of the file, you will find examples that you can edit to fit your network topology.

For each peripheral node that is expected to request a boot load from the central site server, an entry must be made into the `bootptab` file. The entry for the sample topology shown in Figure 313 contains the following information:

```
remote:ip=129.213.201.22:hp=1:sm=255.255.255.0:\
:hd=config-directory:bf=boot.68k:bh=129.213.201.21:\
:hh=frame:ci=30:fs=129.213.201.24
```



The `hd` tag points to the directory where the configuration files are located. If this tag is used, the specified value overrides whatever defaults may be specified elsewhere.

In the case where a non-3Com BootP server is used, the tags `bh`, `hh`, `ci`, `ph`, and `sm` will not work. Instead, the MAC address of the port on the peripheral node over which autostartup is being executed should be specified for each entry in the `bootptab` file. These entries contain the following information:

```
remote: ip=129.213.201.22:hd=config-directory:\
:bf=boot.68k:ht=ethernet:ha=080020011380:\
fs=129.213.201.24:
```

where:

| | |
|----------------------------------|--|
| <code>remote</code> | is the name of the remote bridge/router. |
| <code>ip=129.213.201.22</code> | is the IP address that is assigned to the system named "remote." |
| <code>hp=1</code> | is the number of the central node port that is connected to the peripheral node. (The value assigned to <code>hp</code> can be alpha-numeric. When using an alphabetic character, it should be entered in all capitals.) |
| <code>sm=255.255.255.0</code> | is the subnet mask. |
| <code>hd=config directory</code> | is the pathname of the home directory on the TFTP server where the configuration files exist. |
| <code>bf=boot.68k</code> | is the name of the boot file. |
| <code>bh=129.213.201.21</code> | is the IP address of the port on the central node over which the peripheral node sends the BootP request. |
| <code>hh=frame</code> | is the interface type of the central node port that is connected to the peripheral node. For a PPP link, specify <code>hh=ppp</code> . |
| <code>ci=30</code> | is the connection ID number (assigned by the Frame Relay service vendor). You do not need to specify this tag for a PPP link. |
| <code>fs=129.213.201.24</code> | is the IP address of the TFTP server. |
| <code>ht=ethernet</code> | is the hardware type for specifying the type of MAC, Frame Relay, or PPP address of the peripheral node that is executing the autostartup. |

`ha=08002001138` is the MAC address of the port over which autostartup is being executed.

- b Install the 3Com Upgrade Utilities on the Sun, HP, AIX, or Wintel PC network management station.



You can use your network management station as the BootP and TFTP server or you can set up these functions on an additional server on the network.

These utilities are provided by 3Com on CD-ROM and contain the 3Com implementation of the BootPd program, which is defined in RFC 951 and RFC 1048. When BootPd starts, it reads its configuration file `/etc/boottab`, then sends a BOOTREPLY packet based on the contents of the `/etc/boottab` file for a BOOTREQUEST.

The distribution CD-ROM contains an installation script and UNIX manual pages to document the command line syntax of the utilities. For more information about the Upgrade Utilities, see *Upgrading Enterprise OS Software*.

To set up the BootP server on a UNIX network management station, enter:

```
bcmsetup -BootP
```

To set up the BootP server on a PC (Windows) network management station, change directories to `\usr\3Com\bcm\bin` and enter:

```
BootP
```

- 3 Set up the TFTP server.

The file server (IP address) pointed to by the `fs` tag in the `BootPtab` file must have a TFTP server mechanism. Any UNIX-based operating system supports this requirement. TFTP services can also be provided by other network operating systems.

- a Create the configuration file directory on the TFTP server.

You can create any directory as long as your BootP server can support the Root Path option.

- When you use the 3Com Upgrade Utilities for your BootP server, create the directory that is specified in the "hd" parameter in the `/etc/BootPtab` file.
When the directory is specified in the "hd" tag, the configuration directory on the TFTP server would be `/root_directory/config-directory/`, where "root_directory" is the path to the host image files grandparent directory (or `tftpboot` in most cases).
- When using IP addressing to identify remote nodes, configuration file directory could also be `/tftpboot/autostartup/<ip-address-of-the-remote-node>`. In our example that directory would be `/tftpboot/autostartup/129.213.201.21`. (In this case it is not necessary to specify the "hd" tag in the `/etc/bootptab` file.)
- When using MAC addressing, create a directory with the pathname as `root_directory/CLIENTS/<MAC address>`, where "root_directory" is the path of the host image files' grandparent directory and `<MAC address>` is the MAC address of the device.

BootP first tries IP-based addressing to retrieve configuration files from the BootP server however, if IP-based addressing fails after one minute, it tries MAC-based addressing.



For the TFTP server, the pathname is case-sensitive. The directory *CLIENTS* must be uppercase. System-dependent path separators // or \ are both acceptable characters.

- b** Create configuration files for each device you want to autostartup. Configuration files can be one of two types of files: a single ASCII text file or a set of encoded configuration files specific to the device being autostarted.

The most convenient and manageable method of generating a configuration file is to create an ASCII text configuration file using a text editor.

This ASCII text configuration file contains ASCII text string of the commands required to configure the services that the device will need. Since this is an ASCII text file, it can be easily modified as necessary. See the LoadConfigs command described in the Commands chapter in *Reference for Enterprise OS Software* for information on the contents of the ASCII text file. The ASCII file must be named boot.cfg.

You can also use encoded configuration files called compact configuration store (CCS) files. Creating CCS files requires that you use an unconfigured device of the same type (and hardware configuration) as the device to be autostarted. You must enter the required configuration commands for the device to be autostarted on the unconfigured device as if it were the device to be autostarted. This process creates the necessary CCS files. Because the CCS files are not text files, they are not directly editable.

- c** Copy the configuration files that you just created from the system you used to create them to the configuration directory you created in step a.
- d** Create the CONFFILE file.

Using a text editor, create an ASCII text file named CONFFILE in the configuration file directory on the TFTP server. One CONFFILE needs to be created for each device to be autostarted.

CONFFILE is a text file that contains configuration filenames. This file must contain the filenames of all the configuration files that the TFTP server provides for the device to be autostarted. For example, a CONFFILE can contain the following contents:

```
ip<sep> iprip<sep>rtmnet<sep>system<sep>
```

Where IP, IPRIP, RTMNET, and System are the names of the CCS files contained in the same directory and <sep> is the separator of each file. The <sep> can be a blank, a tab, a form feed, a carriage return, or a new-line character.

CONFFILE can also contain the filename boot.cfg where boot.cfg is an ASCII file containing a list of ASCII commands that are executable on the bridge/router.

- 4** When all the required configuration files and services are in place at the central site, you are ready to set up the remote sites.

For information on installing and cabling the remote node, see the documentation that accompanies the hardware.

- 5** Plug in remote node device.

The device starts up. The initiation of the autostartup phase 2 process depends on the following line types used for the physical link:

- When a leased line is used, plug in the appropriate cables between the bridge/router and the modem to which the leased line is connected.

- When a dial-up line is used, configure the modem or channel service unit/digital service unit (CSU/DSU) to dial the central node so that a physical link can be established.

Autostarting the Central Site Node

You can autostartup the central site node. To do so, set up the central site node configuration file on the TFTP server and BootP mapping the same as for the remote nodes.

Reset the central site node. When the device attempts to load configuration files and finds that none exist, the system sends a BootP request trying each port until it receives a response from an external BootP server.



When autostarting the central site node bridge/router, MAC to IP addressing must be used in the BootPtab file. You cannot autostart a NETBuilder II bridge/router with a DPE module or a PathBuilder S5xx series switch when it is used as a central site node.

The BootP server provides the central site device with the address of the TFTP server that has its configuration file. The device then sends a TFTP request for a load of its configuration file. When the configuration file is received the node loads its configuration. If the configuration files are CCS files, the node reboots in order for CCS configuration to take place. If the configuration file is an ASCII (boot.cfg) file, the node does not reboot but executes the Enterprise OS commands listed in the file and enters its normal operating state.

How Autostartup Works

The Autostartup feature is a two-phase process. This section describes what happens during each phase.

Autostartup Phase 1

Phase 1 of the Autostartup process begins when the bridge/router is plugged in.

During phase 1, a bridge/router automatically detects certain local and wide area port and path attributes. Table 79 lists the detected attributes.

Table 79 Detected Peripheral Node Attributes

| Local Area Path Attribute Detected | Wide Area Path and Port Attributes Detected |
|------------------------------------|--|
| Token ring speed* | WAN protocol (PPP or Frame Relay) that runs on a port
Line type, for example, leased or dial-up
DTE connector type that you have cabled† |

* Applies to model 32x and 52x bridge/routers only.

† Applies to model 42x bridge/routers only

Phase 1 for model 42x bridge/routers detects the path attributes instantly on initial system startup; however, if the connector is changed during normal operation, it can take several minutes for the auto detection software to sense the connector. For more information, see “Automatic Attribute Detection for DTE Ports on Remote Bridge/Routers” next.

After the attributes listed in Table 79 are detected, the bridge/router establishes a physical link and then a data link between itself and the central node. Phase 1 is

complete, and phase 2 of the autostartup process begins. For more information, see “Autostartup Phase 2” later in this chapter.

Automatic Attribute Detection for DTE Ports on Remote Bridge/Routers

For model 42x SuperStack II bridge/routers, the autostartup phase 1 process also detects the DTE port you have cabled. This process only works only on DTE ports and only when the -PORT OWNER parameter is set to AUTO (default). Autostartup can take several (three to five) minutes if the cable is changed during normal system operation.

When establishing the physical and data links between themselves and the central node, model 42x bridge/routers attempt to detect the connector type, owner, and line type of the path associated with the cabled DTE port. The remote router detects these path characteristics by first attempting to try connector and line type combinations. The connector and line types are tried in the following order:

- RS-232 with leased line
- RS-232 with dial-up line
- RS-449/V.36 with leased line
- RS-449/V.36 with dial-up line
- V.35 with leased line
- V.35 with dial-up line

The bridge/router scans each line quickly to determine the connector type most likely being used. After it successfully detects the connector and line types, the bridge/router tries to detect the owner using a similar process. The scanning process continues periodically so that connector changes can be quickly determined.

The possible owners are tried in the following order:

- PPP
- Frame Relay

Knowing the order in which the connector and line types and owners are tried can help you anticipate how long the establishment of the physical and data links will take. For example, the detection of a V.36 dial-up line running Frame Relay will take longer than the detection of an RS-232 dial-up line running Frame Relay because the V.36 dial-up line is tried later than an RS-232 dial-up line.

To determine the progress of the establishment of the physical and data links, follow these steps:

1 Enter:

SHow -PORT DIAGnostics

A display shows you which connector type, port owner, and line type has been tried and deemed a failure, and which is currently being tried.

2 Look at the LEDs on model 42x bridge/routers associated with the DTE connectors to determine which connector is currently being tried.



Because of a signal irregularity in the RS-449 connector, the autostartup detection feature occasionally reports that the RS-449 connector is a V.35 connector. This condition eventually corrects itself.

Autostartup Phase 2 During phase 2 of the autostartup process, the peripheral node obtains necessary configuration information from central site servers across the PPP or Frame Relay line. The peripheral node obtains information including:

- IP address
- Boot file location
- Configuration files

For phase 2 to work, you must configure the UDP Broadcast Helper feature on the central node and you must configure two servers on the central site network: a BootP server and a TFTP server. The BootP server “listens” for BootP requests and forwards an IP address and boot file location information to the requesting node. The TFTP server forwards configuration files to the remote node.

During phase 2:

- The remote node bridge/router broadcasts a BootP request packet to the central node.
- The central node forwards the BootP request to the BootP server on the central site network using the UDP Broadcast Helper feature.
- The BootP server replies to the broadcast BootP request packet. This reply contains IP addresses for the remote node bridge/router and TFTP server and the location of the appropriate configuration files.
- The remote node bridge/router sends read request packets that request certain configuration files from the TFTP server on the central site network.
- When the TFTP server receives the read request packet, it begins to transfer the requested configuration files to the remote node bridge/router. The file transfer proceeds as a series of transfers and acknowledgments until the file transfer is complete.
- When the file transfer is complete, the remote node bridge/router automatically reboot and applies the newly acquired configuration files. When ASCII configuration files are used, no reboot is done.

Sample Configurations

These sample configurations not only illustrate many of the basic concepts and enhancements discussed earlier in this chapter, but they also provide solutions for working with with a Cisco router on the central site.

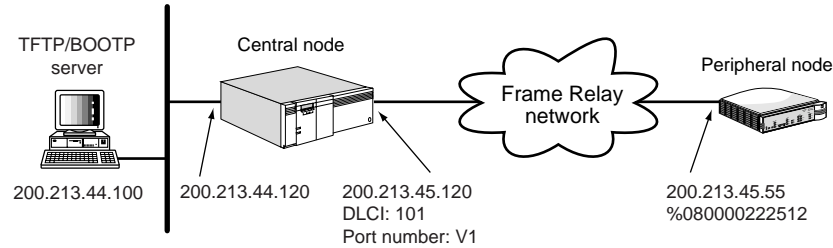
The WAN link can be either PPP or frame relay, using either a 3Com or Cisco central router. The BootP and TFTP server can be either 3Com proprietary or any other standard BootP server. (If the remote node is a nonproprietary BootP server, it must be identified by MAC address.) For more information on BootP and TFTP servers, see *Upgrading Enterprise OS Software*.

BootP Server 3Com Nonproprietary BootP Servers

In other BootP servers, the identifying tag can only be the Ethernet address. This is the WAN port MAC address of the peripheral router over which the BootP request pdu travelled to the central router.

Sample Configuration: Frame Relay WAN This sample setup demonstrates a typical configuration for a 3Com central node, 3Com proprietary BootP server, and frame relay WAN, as illustrated in Figure 315.

Figure 315 Autostartup with a Central Node, 3Com BootP Server, and Frame Relay WAN



First, plug in the peripheral router.

At the central router, follow these steps:

- 1 Set an IP address for the Ethernet port, using the IP address 200.213.44.120.
- 2 Set an IP address for the WAN port, using the IP address 200.213.45.120 .
- 3 Determine the DLCI to which the peripheral router needs to talk, in this case 101.
- 4 Determine the port to which the DLCI is linked, in this case V1.
- 5 Enable IP routing.
- 6 Set up UDPHELP with 67 as the active port and the BootP server as the forward address, using 200.213.44.100 as the address.

At the BootP server, follow these steps:

- 1 Edit the /etc/BootPtab file, entering the following information:
Router1:ip=200.213.45.55:sm=255.255.255.0:hh=frame:hp=V1:
bh=200.213.45.120:ci=101:fs=200.213.44.100:gw=200.213.45.120:

In the case of a 3Com nonproprietary BootP server, edit the etc/BootPtab file by entering the following information:

- 1 **Router1:ht=ethernet:ha=080000222512:ip=200.213.45.55:fs=200.213.44.100:**
- 2 The BootP reply from the BootP server is sent to the gateway (200.213.45.120), so set 200.213.44.120 as the gateway for the BootP server for this address.

At the TFTP Server, follow these steps:

- 1 Create the directory /tftpboot/autostartup/200.213.45.55. (tftpboot is normally the root directory for TFTP.)
- 2 Create the file CONFFILE in this directory. This file needs to have a single entry in it: boot.cfg.
- 3 Create the boot.cfg file.

This is an ASCII file with all the configuration commands in it. You could have the last command as rename command, so that the next time the router boots it will not execute the commands in boot.cfg again. A file is created in the configuration directory to log the execution of the commands in the boot.cfg file. In the case

that config.log already exists, it is renamed as config.bak and a new config.log file is created.

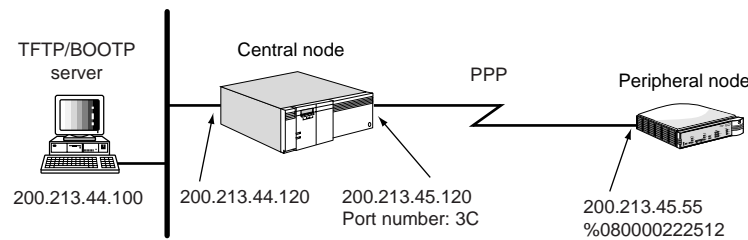


capture.cfg will not capture the commands executed in the boot.cfg file (ASCII boot).

Sample Configuration: PPP WAN

This sample setup demonstrates a typical configuration for a 3Com central node, 3Com proprietary BootP server, and PPP WAN, as illustrated in Figure 316.

Figure 316 Autostartup with 3Com Central Node and BootP Server and PPP WAN



First, plug in the the peripheral node.

At the central site, follow these steps:

- 1 Enable PPP on the WAN.
- 2 Set up UDPHELP with 67 as the active port and the BootP server as the forward address, using 200.213.44.100 as the address.

At the BootP server, follow these steps:

- 1 Edit the /etc/BootPtab file, entering the following information:

```
Router2:ip=200.213.45.55:sm=255.255.255.0:hh=ppp:hp=3C:\
bh=200.213.45.120:fs=200.213.44.100:gw=200.213.45.120:
```

In the case of a 3Com a nonproprietary BootP server, edit the /etc/BootPtab file by entering the following information:

```
Router1:ht=ethernet:ha=080000222512:ip=200.213.45.55:fs=200.213.44.100:
```

- 2 The BootP reply from the BootP server is sent to the gateway (200.213.45.120), so set 200.213.44.120 as the gateway for the BootP server for this address.

At the TFTP server, follow these steps:

- 1 Create the directory /tftpboot/autostartup/200.213.45.55. (tftpboot is normally the root directory for TFTP.)
- 2 Create the file CONFFILE in this directory. This file needs to have a single entry in it: boot.cfg.
- 3 Create the boot.cfg file.

This is an ASCII file with all the configuration commands in it. You could have the last command as rename command, so that the next time the router boots it will not execute the commands in boot.cfg again. A file is created in the configuration

directory to log the execution of the commands in the boot.cfg file. In the case that config.log already exists, it is renamed as config.bak and a new config.log file is created.



capture.cfg will not capture the commands executed in the boot.cfg file (ASCII boot).

Cisco Router at the Central Site

This section provides work-around configuration information if you are experiencing interoperability issues with a Cisco router.

For a PPP WAN, configure the Cisco router by entering the following information:

```
Serial0
ip address 200.213.45.120 255.255.255.0
ip helper-address 200.213.44.100
encapsulation ppp
```

For a Frame Relay WAN, configure the Cisco router by entering the following information:

```
Serial0
ip address 200.213.45.120 255.255.255.0
ip helper-address 200.213.44.100
encapsulation frame-relay ietf
```



ietf is used for the RFC standard encapsulation.

```
frame-relay map ip 200.213.45.55 101 ietf
```



Unlike using a 3Com router at the central site, a static entry has to be entered for the DLCI if the BootP response pdu should reach the peripheral router.

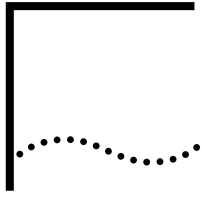
Alternately, you can configure subinterfaces on the WAN port to handle more than one peripheral router connected to the WAN. At the BootP server (3Com proprietary server only), the gateway tag (gw) can be used to identify the entry instead of the MAC address. Enter:

```
router2 : ht = ethernet : ip = 200.213.45.55 : fs = 200.213.44.100 : gw =
200.213.45.120
```



The gateway entry has to be unique for each entry, unlike using a 3Com router at the central site where the bh entry can be the same but differentiated by the DLCI number.





CONFIGURING WITH ASCII FILES

This chapter describes configuring a device with ASCII files.

- For information on using the ASCII boot feature, see “ASCII Boot” on page 896.
- For information on using the ASCII capture feature, see “ASCII Capture” on page 902.

Overview of ASCII File Usage

When a NETBuilder bridge/router or PathBuilder S5xx series switch, called the device in this chapter, is configured, encoded configuration files are created automatically and stored on the device’s local storage. These encoded configuration files are automatically updated as additional configuration changes are made. If the device should reboot, these encoded configuration files are read to ensure that the device comes up with the same configuration as before the reboot.

When the device is upgraded to a new software version, these encoded configuration files need to be upgraded for the new release using 3Com’s Upgrade Management Utility programs. Because these configuration files are in an encoded format, you cannot easily edit them to make changes. In addition, should you want to configure another similar device, you would typically have to reenter all of the configuration commands at the new device.

ASCII Text Configuration Files

You can now avoid this inconvenience by maintaining all or most of the configuration for a device in an ASCII text configuration file. This ASCII text file can be created with any simple text editor on a PC or UNIX workstation. It contains the configuration commands for a specific device

Downloading the ASCII Text Configuration File

To download the ASCII text configuration file to the device, you must use the COpY (TFTP file transfer) or GET (FTP file transfer) command at the device.



Some minimal configuration of the device is needed before the file transfer operation can occur.

The configuration commands for the various services are documented in the *Reference for Enterprise OS Software*. In addition, see *New Installation for Enterprise OS Software*, which explains how to boot and set up the NETBuilder bridge/router. See *Using the PathBuilder Switch* for information on starting up the PathBuilder S5xx series switch.

Executing the Configuration File with LoadConfigs

After the ASCII file has been downloaded to the device, you can execute the LoadConfigs command at the device to execute all of the configuration commands in the ASCII text configuration file. When LoadConfigs is used to execute these configuration commands, the encoded configuration files will still be created or updated, as necessary. The LoadConfigs command is fully described in *Reference for Enterprise OS Software*.

Additional Configuration File Uses

This process can then be used to configure additional device by simply editing the ASCII text configuration file to generate configuration commands unique for each device. The minimal configuration commands that were needed to enable the file transfer can also be included in the ASCII text configuration file.

ASCII Boot

You would do this to take advantage of the ASCII Boot feature, which is a special extension of the LoadConfigs command. The ASCII Boot feature allows you to execute the equivalent of a LoadConfigs command automatically when the device is booted deleting its previous configuration. It is called ASCII Boot because you are booting your bridge/router with an ASCII file that will be used to configure the bridge/router. For more information about ASCII Boot, see “ASCII Boot” later in this chapter.

AutoStartup

It is also possible to boot an unconfigured device and load it with its configuration without ever having to enter a single configuration command at the device. This is the AutoStartup feature which is described in the Configuring Autostartup chapter.



Autostartup is currently not supported on the NETBuilder DPE bridge/router and PathBuilder S5xx series switch.

ASCII Capture

After the device is booted, you may decide to make additional configuration changes. All configuration changes, with the exception of the UserManager command, are captured in an ASCII text file in the configuration directory. The configuration change are captured in a UI command format that can be executed by LoadConfigs. Configuration commands are captured regardless of how they were originally executed, that is, through Web Link, menu interface, macro, UI command, REMote command, scheduler, or SNMP SET requests.

The configuration of passwords and other secure parameters are also captured, but not with the value you entered. This is because the capture file is a clear text file and for security reasons, the password/secure parameter values are not captured in clear text. For more information about ASCII Capture, see “ASCII Capture” later in this chapter.

ASCII Boot

The ASCII Boot feature provides a way for you to configure and maintain the complete configuration of a device in a single ASCII text configuration file. There are some minor limitations in the ASCII Boot feature, which prevent it from being

used to store a small subset of configuration commands. Those limitations are explained later in this chapter.

Creating the ASCII Text File

To use the ASCII boot feature, you create an ASCII text file offline that contains all of the UI configuration commands needed to configure the device. This ASCII text file must follow the same rules as any file executed by the LoadConfigs command. These rules are as follows:

- A line is zero or more ASCII characters terminated by a carriage return, line feed, carriage return followed by line feed, or line feed followed by carriage return.
- Only one command is allowed per line.
- Only the following commands are allowed: SETDefault, ADD, DEL, SHow, SHowDefault, ReName, and SAVEbgp.
- Comments, which are lines with the first character as #, and blank lines are allowed.

Downloading the ASCII File to the Device

Next you transfer this text file to the device and store it in the configuration directory with the filename boot.cfg. This is done by executing the TFTP COpy or the FTP GET command on the device, or it can be done by executing bcmcp or bcmftp at a network management station that has the 3Com Upgrade Management Utilities programs installed. After the file is transferred to the device, the device can then be rebooted.

When the device comes up, any pre-existing configuration files in the configuration directory are deleted. The messages "Deleting existing configuration files" and "Completed deleting configuration files" is displayed before and after the existing configuration files are deleted.

Executing boot.cfg

Next, all of the configuration commands in the boot.cfg file are executed. Functionally this is similar to booting a device with no configuration and then executing all the configuration commands needed to configure the device, except that existing macros are not deleted and SysconF changes remain in effect.

To indicate that the commands in the boot.cfg file are executing, the following messages are displayed before and after the commands in the boot.cfg file are executed:

- Executing configuration commands in BOOT.CFG
- Completed executing configuration commands in BOOT.CFG .

If the router is a QuickStep VPN router, a set of configuration commands are executed before the configuration commands in the boot.cfg file are executed. These QuickStep VPN configuration commands are listed in the first part of the Configuring Quick Step VPN chapter.

If the device has intelligent I/O modules (6 port Ethernet, ATM module, Multiport BRI modules), these modules are loaded before any of the commands in the boot.cfg file are executed. The "System Initialized and Running" message is also displayed until all of the commands in the boot.cfg file are executed. No user

interaction is possible until after all of the commands in the boot.cfg file have executed.

Also, the encoded configuration files are created. However, as with executing the LoadConfigs command, the configuration files are cached in RAM first and are usually not written to nonvolatile flash memory until after all of the commands in the boot.cfg file have been executed.

The configuration commands in the boot.cfg file are not displayed as they are executed, unless the Enterprise OS InterAction parameter has been configured in the boot.cfg file to enable the display of LoadConfigs commands. See the Commands chapter in *Reference for Enterprise OS Software* for more information about the InterAction parameter.

The CONFIG.LOG File

A log file CONFIG.LOG is created in the configuration directory to log the execution of the commands in the boot.cfg file. If a configuration command does not execute successfully or it is a command not supported by LoadConfigs, execution does not stop, but continues with the next command. The commands are written to the log file with a prefix that is the line number of the command in the boot.cfg file. As with the normal LoadConfigs operation, comments and blank lines are also written to the log file. When the log file CONFIG.LOG already exists, it is renamed CONFIG.BAK. When the file CONFIG.BAK already exists, it is deleted before the rename operation occurs.

Renaming the boot.cfg File

After all of the configuration commands have executed and the software is operational, you may choose to rename the boot.cfg file. Since encoded configuration files have been created, these files can be used the next time the router is booted instead of deleting all of these encoded files and executing all of the configuration commands. You can do this by making the last command in the boot.cfg file be a ReName command to rename the boot.cfg file to a different filename.

If the ASCII Boot feature has been executed, an additional status message is displayed in the SysconF Boot Statistics display. This message shows the number of commands in the boot.cfg file that failed to execute successfully.



WARNING: Because the ASCII text file is a clear text file, you should be careful to not configure passwords or other secure parameters with this feature, unless you are confident of the security of the boot.cfg file.

Limitations of ASCII Boot Feature

By default, any additional configuration changes that are made to the device after it is operation are not added in the ASCII Boot file (boot.cfg). This means that if the device is rebooted and the ASCII Boot feature is invoked again, the additional configurations changes will not be in effect. However, the additional configuration changes are automatically captured to the ASCII text file CAPTURE.CFG in the configuration directory. Regardless of how the configuration of the device was accomplished, any UI command, that can be executed by LoadConfigs, is written to the capture file. In addition, you can set up the ASCII Capture feature to automatically append all additional configuration changes to the end of the

boot.cfg file by configuring the SYS services CAPTURE parameter to the appropriate value.

Because the ASCII boot feature is an extension of LoadConfigs, it has the same command set limitations as LoadConfigs. The ASCII boot feature has the following limitations:

- Configuration commands that require user interaction are not supported. Thus, the AddUser, PassWord, and DiscoverRoute commands are not supported by the ASCII Boot feature and LoadConfigs.
- Macros cannot be read or executed in the configuration file. Non-configuration commands are not supported in the ASCII Boot file.
- Menu-driven configuration commands are not supported in the boot.cfg file. The menu driven commands are the InStall, SysconF, SysInfo, SysPassWord, and UserManage commands.
 - The InStall command is a command available only on the Boundary Router package and it provides a simpler way to make some basic configuration changes for which there are equivalent UI commands supported by LoadConfigs to do the same thing.
 - The SysconF configuration changes are not stored in configuration files and are thus not affected by the deletion of the existing configuration files. They will always remain in effect regardless of whether the ASCII boot feature is used.
 - There are SYS services parameters supported by LoadConfigs that can be configured to perform the equivalent function of the SysPassWord command.
 - The UserManage command is not supported at this time.

Basic Configuration Procedure

The ASCII boot feature requires some initial setup. To set up the ASCII boot feature, follow these steps:

- 1 Using a simple text editor at your network management station, create a file containing the UI configuration commands to be executed. (Do not create the ASCII text command file with a word processor that embeds text and format processing command codes in the file.) Since LoadConfigs will actually be executing the ASCII text file, the file must have the format prescribed by LoadConfigs. See the LoadConfigs command in the Commands chapter in *Reference for Enterprise OS Software* for specific LoadConfigs format guidelines.

The following is an example of an ASCII boot text file:

```
***** SYS configuration *****
SETD -SYS NMPrompt = "R-205
SETD -SYS SysNAME = "ROUTER-205
SETD -SYS WelcomeString = "Router 5 in bldg 200"
***** IP configuration *****
SETD !1 -IP NETaddr=101.101.101.101 255.255.255.0
SETD -IP CONTROL = (ROUTE, SECURITY)
SETD -IPSec CONTROL = (EXTENDED, LABELADD)
SETD -IPSec FileServer=Yes
***** Firewall configuration *****
```

```
SETD !1 -FireWall CONTrol=Filter
SETD !1 -FireWall DefAction=Log
ADD !1 -FireWall FTPIn Permit
ADD !1 -FireWall FTPOut Permit
```

- 2 Boot the device, if it is not already operational. See the appropriate platform guide for the hardware being booted, if you are unfamiliar with how to boot the device.
- 3 Configure the device for IP routing, if this has not already been done. If you are unfamiliar with how to configure the router for IP routing, see the Configuring IP Routing chapter.



The commands needed to configure the router for IP routing must be included in the ASCII boot text command file.

- 4 Use the PING command to verify that you have connectivity between the network management station and the device. This should be tried at both the device and the network management station. See *Reference for Enterprise OS Software* if you are unfamiliar with the Enterprise OS PING command.
- 5 At this point, the rest of the ASCII boot configuration file setup can be done in one of two ways. If the user wants to execute every step of the process, he can do the following:
 - a To set the default directory on the router for file transfer commands to the configuration directory, enter:

ChangeDir

- To transfer the file using TFTP, use:

```
COPY <TFTP server IP address>:<path>/<filename> boot.cfg
```

- To transfer the file using FTP, enter:

```
GET <FTP server IP address>:<path>/<filename> boot.cfg
```

Before you can use FTP to transfer the file, you will need to have set up the FTP username and password for the router FTP client to use to connect to the FTP server. The username and password are set up through the Sysconf menu command. See *Reference for Enterprise OS Software* if you are unfamiliar with this command.

For the ASCII boot feature to work, the name of the ASCII text command file on the router must be boot.cfg and the file must be in the configuration file directory. Filenames on the bridge/router are not case-sensitive.

- Use the LoadConfigs command to test the ASCII text command file, by entering the following command:

LoadConfigs boot.cfg

LoadConfigs will attempt to execute all of the UI commands in the boot.cfg file. If any of the configuration commands fail, LoadConfigs will terminate with the following error message "Error - Configuration command failed."

The commands executed and the output generated by executing the commands will be displayed on the console and also written to the log file CONFIG.LOG in the configuration file directory. The log file will be written in the standard UNIX format, which means that each line of text in the log file is terminated with only a NewLine (also called LineFeed) character.

The following is an example of a log file generated by executing the previous example of an ASCII with the IE option:

```
LoadConfigs executed Thu Mar 19 14:22:12 1998 via UI command
[1]LC: # ***** SYS configuration *****
[2]LC: SETD -SYS NMPrompt = "R-205 #"
[3]LC: SETD -SYS SysNAME = "ROUTER- 205"
[4]LC: SETD -SYS WelcomeString = "Router 5 in bldg 200"
[5]LC: # ***** IP configuration*****
[6]LC: SETD !1 -IP NETaddr=101.101.101.101 255.255.255.0
[7]LC: SETD -IP CONTROL = (ROute, SECurity)
[8]LC: SETD -IP SecCONTROL = (EXTended, LabelAdd)
!<portlist> required for SecCONTROL
The command did not execute successfully
[9]LC: SETD -IPSec FileServer=Yes
[10]LC: # ***** Firewall configuration *****
[11]LC: SETD !1 -FireWall CONTROL=Filter
[12]LC: SETD !1 -FireWall DefAction=Log
[13]LC: ADD !1 -FireWall FTPInPermit
[14]LC: ADD !1 -FireWall FTPOutPermit
Warning - The following 1 configuration command(s) failed
8
```

The EnterpriseOS CAT command can be used to view the contents of the log file. If a configuration command fails, it has the following diagnostic message after the command, as seen in the above example.

The command did not execute successfully

The end of the log file also has a warning message indicating the number of commands that failed and the line numbers of the first 32 commands that failed and the message "LoadConfigs terminated due to UI command failure" indicating LoadConfigs terminated prematurely without executing all of the commands.

LoadConfigs has an IgnoreErrors option to continue executing even when commands fail to execute successfully. When that option is selected, the message "LoadConfigs terminated due to UI command failure" is not written to the log file as shown in the example above. The ASCII boot feature initiates the LoadConfigs operation with the IgnoreErrors option. Thus, all of the configuration commands in the boot.cfg file are executed, even if some of the configuration commands fail.

If LoadConfigs is able to execute all of the commands successfully, you are done. If not, you may want to edit the ASCII text command file at the network management station and repeat these steps.



Depending on what commands successfully executed the first time, it is possible for a LoadConfigs to encounter an error, when it executes the same command a second time.

There is an option in LoadConfigs to specify the line number of the first command you want LoadConfigs to execute, so when you repeat these steps, you can skip the configuration commands that executed successfully the first time. In both the console display and log file, the commands are displayed with its line number as in the above example. For more information on the LoadConfigs command, see *Reference for Enterprise OS Software*.

- b** A simpler way would be to use the Enterprise OS Upgrade Management Utility program `bcmloadconfig`. This requires that you install the Enterprise OS Upgrade Management Utilities programs on the network management station. For information on installing the Enterprise OS Upgrade Management Utilities programs, see *Upgrading Enterprise OS Software*. The device must also be configured for SNMP access. This involves configuring the same community string at both the device and at the network management station. For information on configuring the SNMP community string with read/write access on the router, see *Reference for Enterprise OS Software*. At the network management station, the community string is written to a file.

To verify SNMP access to the router, `bcmdiagnose` should be executed. The execute the Upgrade Management Utility program `bcmloadconfig` via the command line.

ASCII Capture

The ASCII capture feature will make it possible for you to save all of the successful configuration changes into a single file that can be executed by LoadConfigs. The supported commands include `SETDefault`, `ADD`, and `DELeTe`. In addition, configuration changes can be added to the same file as the one used by the ASCII boot feature.



The standard encoded configuration files are also maintained.

This command capture occurs regardless of how the configuration change was originally made. That is, commands are captured whether they are made through LoadConfigs execution, TELnet, MEnu, a Scheduler event, execution of a macro, a command entered through Web Link, or an SNMP SET request. Commands are always captured with their full service and parameter names, even if the user originally used `CurrentServices`, `Aliases`, or short forms of the service names and parameters.

Configuration commands executed through the Install utility and QuickStep VPN executed after initialization are captured. The configuration commands executed during initialization process are never captured by the ASCII Capture feature. So the initial QuickStep VPN configuration commands, which are executed the first time QuickStep VPN router is booted, are never captured. Commands executed by the ASCII Boot feature are never captured.

By default, the captured configuration commands are written to the file `CAPTURE.CFG` in the configuration directory. If the file does not exist, it is created. Unlike the ASCII Boot feature, the capture file is never overwritten, deleted, or renamed. Each captured configuration command is appended to the end of the file.

It is your responsibility to manage this file as far as not letting it get too large and cleaning up redundant and commands that no longer are applicable. You can do this by transferring the file to a network management station, editing the file, and then transferring it back to the device. You can even merge the commands in the capture file with the commands in the Boot file, if that feature is also being used.

When configuration changes are captured, they are not immediately written to the capture file in flash memory. The captured commands are first cached in RAM and written to the capture file when the cache buffer is full, when the capture feature is disabled, when the capture file is switched, or when a minute has

elapsed with no configuration changes. You can also flush the capture buffer in RAM at any time by entering the Enterprise OS SAVECapture command.

SNMP SET requests are saved in a new special SNMP command that can only be executed by LoadConfigs. It is not intended for you to ever create your own SNMP commands, because setting a MIB object to the wrong value can cause unpredictable behavior in your device.

By default, the ASCII capture feature is enabled. You can disable this feature by setting the SYS services CAPTURE parameter to the appropriate value. You can also have the captured commands automatically appended to the ASCII Boot file, by again setting the SYS services CAPTURE parameter to the appropriate value.



WARNING: For security reasons, passwords or other secure parameters will not be captured with their configured value. A noop value is substituted in the captured command for the secure parameter.

Limitations As with the ASCII Boot feature, there are also some limitations to the ASCII Capture feature consisting of the following:

- Nonconfiguration commands including those supported by LoadConfigs (SHow, ReName, SAVEbgp, SHowDefault) are not captured.
- Configuration commands that require user interaction are not captured. Currently, these consists of the AddUser, PassWord, UserManage, and DiscoverRoutes configuration commands.
- Macro creation and execution is not captured into the configuration file. However, the SETDefault, ADD, and DElete commands executed by macros are captured.
- For security reasons, passwords and other secure parameters are not captured with the value that you entered. Instead, the configuration command are captured with 6 or more asterisks as the value for the secure parameter. This value for the secure parameters is treated as a noop value. Whenever, a secure parameter is being configured with this noop value, the existing value of the secure parameter is not changed.

SNMP Command The SNMP command executes a configuration change that was originally executed via a SNMP SET request or from setting a few QuickStep VPN configuration parameters using Web Link. When configuration commands are being captured, SNMP SET requests are captured by converting the SNMP SET requests into an SNMP command and writing that command to the capture file.

The syntax of the SNMP command in the capture file is:

```
SNMP <MIB object id> <MIB object type> <MIB object value>
```



This SNMP command can only be executed by Loadconfigs.

Procedure ASCII Capture feature can be enabled and disabled and can be set to capture the configuration commands into the capture.cfg or boot.cfg (file executed by ASCII Boot) files.

By default, ASCII Capture is enabled and the commands are captured to the capture.cfg file in the configuration directory.

- 1 To control the ASCII Capture feature, use:

```
SETDefault -SYS CAPTure = [ Enable | Disable ] , [ BootCfg | CaptureCfg ]
```

- 2 To disable the ASCII Capture feature, enter:

```
SETDefault -SYS CAPTure = Disable
```

- 3 To reenble the ASCII Capture feature, enter:

```
SETDefault -SYS CAPTure = Enable
```

This command by default starts capturing into the ASCII text file Capture.cfg.

- 4 To have the captured commands automatically appended to the end of the ASCII Boot file, enter the following command:

```
SETDefault -SYS CAPTure = (Enable, BootCfg)
```

- 5 To switch the capture file back to the default capture file, enter the following command:

```
SETDefault -SYStem = CaptureCfg
```

- 6 To flush the command capture buffer in RAM, enter the following command:

```
SAVECapture
```

Flushing the Cache

The configuration commands are initially written to a cache in RAM and after 60 seconds of no additional configuration changes or if the cache buffer is full, the cache is flushed to the appropriate capture file.

To flush the capture file cache, enter:

```
SAVECapture
```

Reviewing the Capture File

The CAT command is used to examine the contents of the capture file.

Example 1

To list a capture.cfg file enter:

```
CAT primary/capture.cfg
```

The resulting display is shown below.

```
# Command capture enabled Tue Aug 11 09:30:08 1998 via software
  initialization
# Product version = SW/NBDPE-DW , 11.2.0.25I
  setd -SYS NMPrompt = "rolls #"
  setd !7 -PORT OWNer = we
  setd !7 -PATH Clock = e
  setd !7 -PORT CONTrol = e
```

Example 2

To list a boot.cfg file, enter:

```
CAT start = 100 mode = ln a:/primary/boot.cg
```

This command lists boot.cfg file along with line numbers from the 100th line onwards as shown below:

```
[ 100]Setd -SYS CAPTure = (e,bc)
```



```
[ 101]Setd !5 -PATH Baud = 4096
[ 102]Add !v1 -PORT VirtualPort ppp
[ 103]Setd !v1 -PORT DialInitState = md
[ 104]Setd !v1 -PORT DialNoList "111" type = we pos = 1
[ 105]Setd !v1 -PORT CONTROL = e
```

How Passwords are Captured

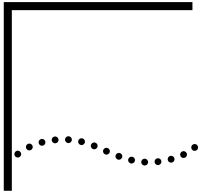
Passwords and other secure parameters are not captured in clear text. The ASCII capture feature is designed to replace password parameter value with a 6 asterisk format and a comment generated above the configuration command notifying you that the subsequent password has not been captured.

In the sample shown below, the password has been captured for the ALU parameter in the PPP services.

Example 3

```
CAT b: /112/boot.cfg
# Command capture enabled Tue Aug 11 09:30:08 1998 via software
  initialization
# Product version = SW/NBDPE-DW , 11.2.0.25I
  setd -SYS NMPrompt = "rolls #"
  setd !7 -PORT OWNeR = we
  setd !7 -PATH Clock = e
  setd !7 -PORT CONTROL = e
  setd !v1 -PORT CONTROL = e
# Secure / Password value in the following command was not captured
  setd !v1 -PPP AuthLocalUser = "*****", "*****"
```





CONFIGURING WIDE AREA NETWORKING USING PPP

This chapter describes how to configure wide area networking using the Point-to-Point Protocol (PPP).

The wide area bridge/router supports PPP for point-to-point communication. PPP is a standard protocol that provides serial line connectivity between two NETBuilder bridge/router or between a NETBuilder bridge/router and a bridge/router built by another vendor running PPP.



For conceptual information about PPP, see "How PPP Works" later in this chapter.

Configuring Point-to-Point Protocol Communication

Only one wide area protocol is allowed to run over one port, regardless of the number of paths assigned to the port. Figure 20 is an example in which only one path has been assigned to one port. In this figure, bridge/router 1 is running PPP over ports 3 and 4. Figure 21 is an example in which two paths have been assigned to one port. In Figure 21, bridge/router 1 is running PPP over port 3, which has paths 3 and 4 assigned to it.

Figure 317 One Path per Port Configuration

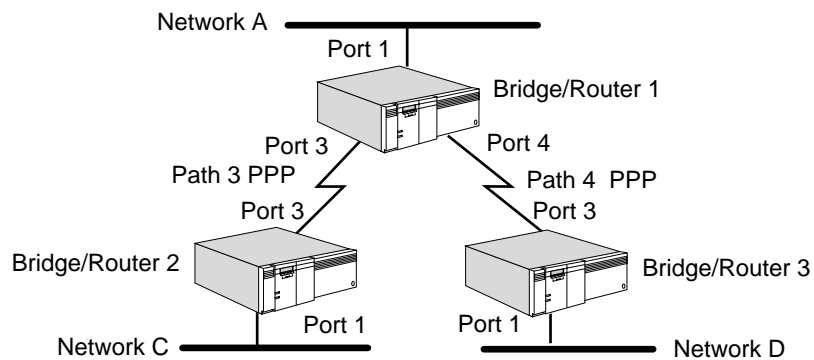
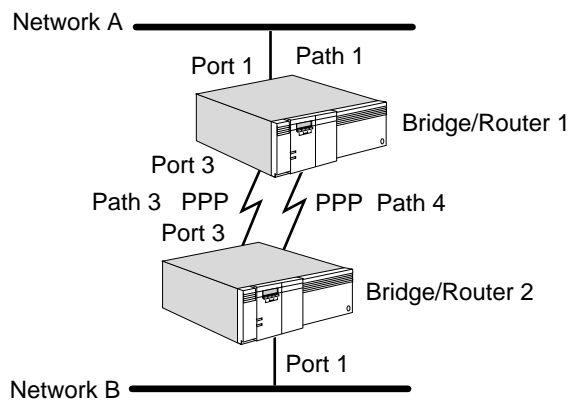


Figure 318 Two Paths per Port Configuration



If you assign multiple paths to one port, as shown in Figure 21, the load sharing feature is enabled. For more information on load sharing, see “Load Sharing and Load Balancing” later in this chapter.

Serial lines running PPP can bridge or route all protocols supported by the NETBuilder bridge/router [Bridging, Transmission Control Protocol/Internet Protocol (TCP/IP), Xerox Network Systems (XNS), open system interconnection (OSI), internetwork packet exchange (IPX), DECnet, AppleTalk, and VINES].

By default, PPP is enabled on serial interfaces.

Enabling PPP

If your bridge/router is built by another vendor, follow that vendor's instructions for enabling PPP.

While enabling PPP, keep in mind the following considerations:

- Before configuring PPP, you must be logged on as Network Manager.
- PPP is the default protocol for serial interfaces on NETBuilder II bridge/routers, and it is automatically enabled. On other platforms, the owner is AUTO.
- When you enable PPP on the NETBuilder bridge/router, you must also enable PPP on the bridge/router at the other end of the serial connection.

If the owner of a port is not PPP, use:

```
SETDefault !<port> -PORT OWNeR = PPP
```

Setting an Authentication Protocol

PPP can be configured to prevent unauthorized access especially for dial-up lines over the Public Switched Telephone Network (PSTN), and also to administer multiple remote users. PPP handles authentication using either the Password Authentication Protocol (PAP), the Challenge Handshake Authentication Protocol (CHAP), Microsoft CHAP (MS-CHAP), or Extensible Authentication Protocol (EAP).

When you are setting up authentication, you must specify a userid and password pair as part of the -PPP AuthLocalUser and AuthRemoteUser parameters. How the pair is used depends on whether PAP, CHAP, or MS-CHAP is configured as the authentication protocol with the AuthProTocol parameter.

With CHAP, you can also specify an interval value in minutes to repeat authentication and to ensure that the identity of a peer has not changed after a link is established. The AuthReptIntvl parameter sets the interval value. Also with MS-CHAP you can use the Microsoft Point-to-Point Encryption protocol (MPPE). By default, in the software packages that include MPPE, 40-bit and 128-bit encryption keys are denied. To enable encryption, you must enable one or both of the encryption keys.

EAP is used to authenticate remote clients only. MD5-Challenge is supported that is identical to CHAP.

Setting Up PAP

To set up PAP with standard bundling, follow these steps:



If you have pre-7.1 software running on the remote bridge/router, you must choose None as the userid when specifying AuthLocalUser and AuthRemoteUser. The password used for both AuthLocalUser and AuthRemoteUser must be the

same as the password used by the remote bridge/router. You must also specify PAP as the AuthProTocol.

On the Local Bridge/Router 1

Using the configuration shown in Figure 21, bridge/router 2 (the remote bridge/router) makes a dial-on-demand call for a user on the network. To set up PAP with standard bundling, on bridge/router 1 follow these steps.

- 1 Specify the AuthLocalUser parameter with a userid and password using:

```
SETDefault !<port> -PPP AuthLocalUser = ["<userid>" | None], "<password>"
```

Remember to enclose the password in double quotes. The userid and password are case-sensitive and can be up to 16 ASCII characters long.



You can set up PAP so that either end of the link may initiate authentication for a session. However, you must have version 7.1 or higher software configured at both ends of the link.

- 2 Configure your remote user identification information.

Both the userid and password must be specified so that multiple remote users have unique user identification information. Do this using:

```
ADD !<port> -PPP AuthRemoteUser ["<userid>" | None], "<password>"
```

- 3 Enable PAP as your authentication protocol using:

```
SETDefault !<port> -PPP AuthProTocol = Pap
```

- 4 Enable all the settings you have specified in the previous steps using:

```
SETDefault !<port> -PPP CONTrol = Enabled
```

On the Remote Router 2

Now, on bridge/router 2, follow these steps:

- 1 Enable the end point discriminator using:

```
SETDefault !<port> -PPP TxEndpointDisc = Enabled
```

When a NETBuilder bridge/router is initiating a call, as it is in this example, the bridge/router sends the end point discriminator.

- 2 Specify the AuthLocalUser parameter with a userid and password using:

```
SETDefault !<port> -PPP AuthLocalUser = ["<userid>" | None], "<password>"
```

Remember that the userid and password must match a userid and password configured on the bridge/router that will receive the call.

- 3 Enable PAP as your authentication protocol using:

```
SETDefault !<port> -PPP AuthProTocol = Pap
```

When bridge/router 2 makes a dial-on-demand call, no SCID is sent. The Config_Req packet includes the endpoint discriminator (ED), which is the MAC address of the port configured to make the call. When bridge/router 1 receives the Config_Req packet request, it queries bridge/router 2 for PAP or CHAP information. Bridge/router 2 responds with the configured userid and password pair. Bridge/router 1 looks up the userid/password pair and binds the link to the specified port.

If additional bandwidth is required, then path 4 comes up connecting to port 3. Path 4 will also include the ED and is bundled with path 3 to create a bundle.

Setting Up CHAP

To set up CHAP, follow these steps. You must always specify a userid.

- 1 Specify the AuthLocalUser parameter with a userid and password using:

```
SETDefault !<port> -PPP AuthLocalUser = "<userid>", "<password>"
```

The password and userid are case-sensitive and can be up to 16 printable ASCII characters long.

- 2 Configure your remote user identification information.

Both the userid and password must be specified so that multiple remote users have unique user identification information. Do this using:

```
ADD !<port> -PPP AuthRemoteUser "<userid>", "<password>"
```

If you are setting up CHAP for remote use, make sure that the remote userid and password pairs are added as AuthRemoteUser entries at the local end, and that the local userid and password pair is added at the remote end.

- 3 Optionally, specify how often CHAP will repeat authentication to verify the identity of the remote user using:

```
SETDefault !<port> -PPP AuthReptIntvl = <minutes> (0-255)
```

If you specify 0, repeat authentication will be disabled.

- 4 Enable CHAP as your authentication protocol using:

```
SETDefault !<port> -PPP AuthProTocol = Chap
```

- 5 Enable all the settings you have specified in the previous steps using:

```
SETDefault !<port> -PORT CONTROL = Enabled
```

Setting Up MS-CHAP

To set up MS-CHAP, follow these steps. You must always specify a userid.

- 1 Specify the AuthLocalUser parameter with a userid and password using:

```
SETDefault !<port> -PPP AuthLocalUser = "<userid>", "<password>"
```

The password and userid are case-sensitive and can be up to 16 printable ASCII characters long.

- 2 Configure your remote user identification information.

Both the userid and password must be specified so that multiple remote users have unique user identification information. Do this using:

```
ADD !<port> -PPP AuthRemoteUser "<userid>", "<password>"
```

If you are setting up MS-CHAP for remote use, make sure that the remote userid and password pairs are added as AuthRemoteUser entries at the local end, and that the local userid and password pair is added at the remote end.



If two-way authentication is needed, make sure that both ends of the connection are configured with the same password to insure proper Microsoft Point-to-Point Encryption protocol (MPPE) operation. It is also important to remember that you must use MS-CHAP if you wish to use MPPE. By default, both 40-bit and 128-bit sessions keys are denied.

- 3 Optionally, specify how often MS-CHAP will repeat authentication to verify the identity of the remote user using:

```
SETDefault !<port> -PPP AuthReptIntvl = <minutes> (0-255)
```

If you specify 0, repeat authentication will be disabled.

- 4 Enable MS-CHAP as your authentication protocol using:

```
SETDefault !<port> -PPP AuthProTocol = MS-Chap
```

- 5 Enable encryption keys to be used by MPPE using:

```
SETDefault !<port> -ppp EncryptCONTRol = (MPPE40 | NoMPPE40, MPPE128 | NoMPPE128)
```

- 6 Enable all the settings you have specified in the previous steps using:

```
SETDefault !<port> -PORT CONTRol = Enabled
```

Verifying Your Configuration

To verify that you have configured PAP, CHAP, MS-CHAP, and MPPE with settings you intended, follow these steps:

- 1 Verify your settings for the AuthLocalUser parameter using:

```
SHowDefault !<port> -PPP AuthLocalUser
```

The AuthLocalUsers are displayed for each port you have configured.

- 2 Verify your settings for AuthRemoteUser using:

```
SHowDefault !<port> -PPP AuthRemoteUser
```

The userids are displayed for remote users you added.

- 3 Verify the protocol that was configured using:

```
SHowDefault !<port> -PPP AuthProtocol
```

The authentication protocol you have configured is displayed.

- 4 Verify the supported MPPE encryption keys using:

```
SHowDefault !<port> -ppp EncryptCONTRol  
tagger
```

Setting Up EAP

EAP can be used for RAS clients only on unbound paths. EAP is an option that is unenabled by default using the PPP Service DefaultAptCtl parameter. When an unbound path begins LCP negotiations, the bridge/router sends a Configure Request with the Authentication Protocol option set to EAP. By default, EAP is set as the bridge/router's most preferred authentication protocol.

The following authentication scenarios are supported:

- EAP MD5-Challenge authentication relay to Radius servers with EAP extension support
- EAP MD5-Challenge authentication relay to Radius servers without EAP extension support

To set up EAP, follow these steps.

- 1 Set up the RADIUS server. (See the Configuring the Remote Access Services chapter .)
- 2 Enable EAP by entering:

```
SETDefault -PPP DefaultAptCtl = EAP
```

Configuring NCPs

The network control protocol (NCP) configuration list is in the PPP component and is system-wide. When the **Enterprise OS** software establishes an LCP connection with a RAS client, it will send out the NCP request packets specified in the NCP configuration list. The NCP configuration list can be configured to contain only NCPs that are supported by the RAS client. This way only configured NCPs will be negotiated by the RAS client.



By specifying the appropriate element in the list, an NCP can be enabled or disabled. If an NCP is disabled, no Configuration Request packet for that NCP is sent and the incoming Configuration Request for that NCP is ignored.

The following configuration request packets are supported:

- IPCP
- SNA-802.2
- SNA
- OSI
- XNS
- IPXCP

To configure NCP, specify the list of negotiable NCPs using:

```
SETDefault -PPP NcpProtocolId = (IPcp, Osi, Xns, SNa-802.2, Sna)
```

Activating LAPB to Reduce Noisy Lines

Normally, when the PPP Protocol is used, the LAPB Service is not required to be active. However, to solve the problem of noisy lines when using these protocols, you need to activate the LAPB Service. The bridge/router has noisy lines if it experiences the following problems:

- A temporary lack of response occurs even though the other end is active.
- LAPB assumes that a line is down, even when it is not, because the clock has gone down temporarily.
- Frames need to be retransmitted frequently.

While LAPB provides a reliable data link, it does add some protocol overhead. Consequently, you need to evaluate the need for LAPB by monitoring the error rates on your lines. To activate LAPB, see the link-level compression procedure in the Configuring Data Compression chapter.

To configure LAPB for noisy lines, follow these steps:

- 1 Increase the T1 parameter to lengthen the amount of time that LAPB waits for an acknowledgment.

This ensures that LAPB does not retransmit a frame unnecessarily.

For example, to increase the amount of time to 4,000 milliseconds on path 3, enter:

```
SETDefault !3 -LAPB T1 = 4000
```

In selecting your value for the T1 parameter be aware that the value you enter is internally divided by 250 milliseconds. As a result, any value you enter less than 250 actually equals zero.

- 2 Increase the N2 parameter.

This increases the maximum number of times a frame is sent after a timeout.

For example, to increase the N2 parameter value to 12 on path 3, enter:

```
SETDefault !3 -LAPB N2 = 12
```

- 3 Decrease the size of the FrameSeq parameter.

When the line is noisy, keep the window size low to keep the number of unacknowledged frames low. In addition, a low window size reduces the number of retransmittals required when the software encounters a corrupted packet.

For example, to decrease the FrameSeq parameter to basic sequencing on path 3 enter:

```
SETDefault !3 -LAPB FrameSeq = Basic
```

How PPP Works

After you have set up and checked the serial lines, PPP performs the following functions:

- Negotiates the maximum size of a packet that can be received over a serial line
- Manages a serial line
- Maintains serial line quality

This section describes the concepts involved in the PPP activities and explains how you can customize PPP operations under different circumstances.

Packet Size Negotiation

During bridge/router startup, the two bridge/routers connected by the serial line negotiate the maximum size of a packet that each can receive. Once the bridge/routers agree on the size of the packet, the negotiation is complete and bridging or routing can begin.

To change the size of the packet, use:

```
SETDefault !<port> -PPP MaxRcvUnit = <bytes>(1-4500)
```

For example, if you want to change the size of the packet from the default of 1,500 (4,500 in the case of token ring) to 1,000 bytes on port 3, enter:

```
SETDefault !3 -PPP MaxRcvUnit = 1000
```

In this example, even if the packet size is negotiated between both bridge/routers to be 1,000 bytes, the bridge/router will continue to receive packets up to 1,500 bytes in size. However, it will discard packets greater than 1,500 bytes in size.

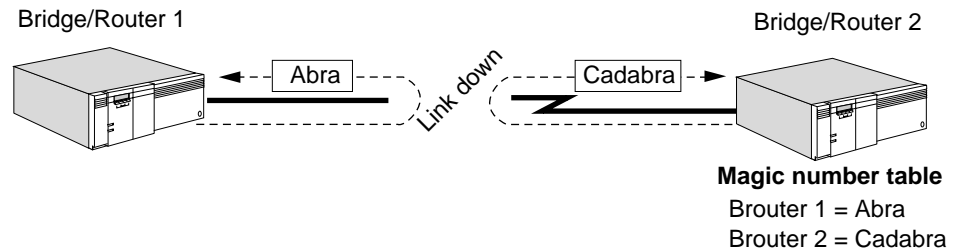
Serial Line Management

A common problem experienced with T1 lines is the loopback of packets. If a T1 line goes down, packets transmitted are looped back as shown in Figure 22. Because the packets are looped back, the bridge/routers may perceive that they are still being sent and received and may not realize that the line is down. To detect a physical loopback problem, each link management packet sent from a bridge/router includes a *magic number* that is checked upon receipt.



The actual time to bring down a line varies depending on the baud rate on the link.

Figure 319 Detecting Loopback of Packets with Magic Numbers



In the configuration shown in Figure 22, imagine that the bridge/router 1 sends out a Link Control Protocol (LCP) packet with a magic number (Abra). When the packet is transmitted to bridge/router2, bridge/router 2 checks the magic number in the received packet against its unique magic number that it has negotiated earlier. If the magic numbers are different, it concludes that a loopback is not occurring.

If upon receipt, bridge/routers 1 and 2 check the magic numbers in their respective returned packets and detect their own magic numbers, both bridge/routers conclude that a loopback is occurring and declare the line down. (When a loopback is detected, the system message "Path <n> loopback" appears on the system console.) After the loopback is removed, the line will come back up. When PPP completes handshaking, the message "Path x is up" appears on the console. This message indicates that PPP is back in the normal state.

The magic number is an option that is negotiated during bridge/router startup. If both bridge/routers at the ends of the serial line connection support this option, then unique numbers are negotiated. These unique numbers are used later in the line quality management packets for loop detection.

Serial Line Quality Maintenance

PPP attempts to maintain the quality of a serial line. If data packets are not received and no echo reply is received in response to echo requests for approximately eight seconds, PPP brings the line down. Once the line goes down, PPP continuously attempts to negotiate until it successfully brings the line up.

How Authentication Works

The Password Authentication Protocol (PAP) uses a two-way handshake method to establish the identity of the peer before a link is established. PAP provides greater security than basic PPP settings. With PAP, a peer determines when the Authentication Request is made. The PeerID/Password pair is sent over the wire in a plain text form.

The CHAP uses a three-way handshake sequence to establish the identity of a peer before establishing a link. It may be repeated at any time if the AuthReptIntvl parameter is set, to ensure that the identity of the peer has not changed. CHAP

relies on the equivalent of userid and password, a “name/secret” pair, to produce a challenge value used for authentication. Because the name/secret pair is never sent directly on the circuit, CHAP provides a higher level of security than PAP authentication. In the CHAP, the authenticator controls when the authentication request is made.

Load Sharing and Load Balancing

When multiple serial links are assigned to a port running PPP, load sharing or load balancing can be used on that port to make more efficient use of the bandwidth of the links.

With load sharing, the fastest link mapped to each port is selected as the primary link, and the other links are considered secondary. When data needs to be sent on that port, it is sent on the primary link until the link is saturated, then the balance of the data is sent over secondary links.

The advantage of load sharing is that if you configure the secondary link for bandwidth-on-demand, it is used only when the data exceeds the bandwidth of the primary link. The disadvantage of load sharing is that it can cause packet misordering, which may be undesirable for some network protocols.

Use load sharing only when data traffic on the port is not sensitive to packet misordering. If the data traffic has a combination of network protocols, some of which are sensitive to packet misordering, and some that are not, you need to select the sequencing feature using the mnemonic filtering scheme to ensure that packets are not misordered for sequencing sensitive protocols. Remember that sequencing works for bridged packets only.

With load balancing, data is split over parallel serial links while preserving sequencing. If the link speeds are the same, the load is split evenly. However, if the link speeds differ, the data is split in proportion to the difference between the speed of the links. For example, if you have two links, and the speed of one is 60% greater than the speed of the other, the faster link receives approximately 60% more data traffic. Through load balancing, all active links can be used to their full capacity.

Load balancing is accomplished only by using the PPP MultiLink Protocol (MLP) (RFC 1717). With MLP, each packet is assigned a sequence number to guarantee in-sequence delivery. In addition, packets may be divided into fragments, which are also assigned sequence numbers.

Packets and packet fragments are sent over the available serial links. The receiving station, also running MLP, reassembles the fragments into packets based on the sequence numbers. Once a packet is completely reassembled, it is released to the client protocol. Packets that cannot be completely reassembled due to lost fragments are discarded.

Whether packets are fragmented depends on their size and the traffic on the links. For example, some packets are too small to benefit from being fragmented. Moreover, if you have only two links, and one of them is saturated, you do not gain a performance advantage by sending packet fragments to this link.

If you use bandwidth-on-demand to back up a single primary link, and you also use load balancing, packets are sent unfragmented on the primary link. As soon as the data traffic exceeds the bandwidth of the primary link, the secondary link is brought up, which in turn enables load balancing. Once the data traffic drops

below a user-specified threshold, the secondary link is brought down and the packets are once again sent unfragmented on the primary link.

If you use bandwidth-on-demand to back up multiple primary links, and you also use load balancing, packets can be fragmented over the multiple primary links. As soon as the data traffic exceeds the bandwidth of all the primary links, the secondary link is brought up, and load balancing is extended to the primary and secondary links. When traffic drops below a specified threshold, the secondary link is brought down and packets are again load-balanced, but only across primary links.

When you use load balancing, paths manually assigned to the same port are referred to as a bundle.

Other ways to create a bundle include:

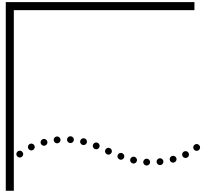
- Dynamically binding paths to a port through dial pooling. This is accomplished by using the SysCallerID port parameter, which allows you to tell the remote router the port number is must assign to the dial pooling path. Remember that you can only use SysCallerID with NETBuilder bridge/routers. You can use dial pooling with either load sharing or load balancing.

For more information about configuring the -SYS SysCallerID parameter, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*. For more information about dial pooling paths, see the Configuring Port Bandwidth Management chapter.

If MLP is turned off (-PPP MlpControl = Disabled), then load sharing is automatically used.

- Standard bundling using Endpoint Discriminator (ED) and authentication. NETBuilder bridge/routers provide class 3, MAC address for the ED. Authentication is achieved through PAP or CHAP if either is configured or negotiated. The bundle ID used by NETBuilder bridge/routers to identify links belonging to the same bundle is defined as a combination of ED and authentication.

For ED support you use the TxEndpointDisc parameter in the PPP service. This parameter indicates whether to send ED for a call on the link.



CONFIGURING WIDE AREA NETWORKING USING ISDN

This chapter describes how to configure the Integrated Services Digital Network (ISDN) interface on model 42x and 52x SuperStack II NETBuilder bridge/routers.

Table 80 lists the steps you must perform to configure the ISDN interface and where to find the information related to each step. 3Com recommends performing these steps in the order in which they are listed.

Table 80 Configuring the ISDN Interface

| Step | Where To Find Information |
|--|---|
| Determine the topology of your ISDN network. | "Planning Your ISDN Network." |
| Determine how you want to use the ISDN interface. | "Deciding How to Use the ISDN Interface." |
| Acquire ISDN services from the telephone company. | <i>WAN Cabling and Connectivity Guide</i> * or the installation guide that you received with your SuperStack II NETBuilder bridge/router. |
| Disable phantom power if necessary. | "Disabling Phantom Power." |
| Configure the paths, ports, and virtual ports (if necessary) associated with the B channels you plan to use. | See the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter, or the installation guide that you received with your SuperStack II NETBuilder bridge/router. |
| Configure the dial-up feature. | See the Configuring Port Bandwidth Management chapter. |
| Configure bridging, routing, and Boundary Routing as desired over the ISDN line. | See the chapters for the bridging and routing protocols you want to configure. See the Configuring Boundary Routing System Architecture chapter for information on Boundary Routing. |
| Configure PPP to run over the ISDN line. | See the Configuring Wide Area Networking Using PPP chapter. |
| Configure the ISDN device at the other end of the ISDN network. | "Setting Up the Remote Device." |

* The *WAN Cabling and Connectivity Guide* can be found on the 3Com Corporation World Wide Web site by entering: <http://www.3com.com/>

ISDN operates at the physical layer of the Open System Interconnection (OSI) Model. Since bridging and all routing protocols and Point-to-Point Protocol (PPP) operate at higher layers of the OSI Model, you can configure these protocols exactly as you would over a LAN or any other type of WAN interface. To configure these protocols to run over an ISDN interface, you do not need to perform additional ISDN-related steps.



For conceptual information, see "How the ISDN Interface Works" later in this chapter.

In this chapter, the term *ISDN interface* refers to the two B channels and the D channel. The term *B channel* refers to a specific B channel. The term *ISDN line* refers to the physical line that connects one ISDN device to another. When the ISDN line is used, it is not assumed that both B channels are being used.

Planning Your ISDN Network

3Com offers the following ISDN systems:

| | |
|---|---|
| Models: | Model 42x and 52x SuperStack II NETBuilder bridge/routers |
| Function: | Boundary router or bridge/router |
| Number and type of ISDN interface: | 1 basic rate interface (BRI) (2B+D) |
| Number of B channels that transmit data: | 2 |

3Com also recommends some BRI terminal adapters (TAs) that allow a non-ISDN bridge/router, such as the NETBuilder II bridge/router, to connect to an ISDN network. For information on accessing TA recommendations, see the *WAN Cabling and Connectivity Guide*. You can find this guide on the 3Com World Wide Web site by entering:

<http://www.3com.com/>

The ISDN systems described above are commonly used in a few different topologies. The first is a Boundary Routing topology; the second is a traditional routed environment where all devices are meshed (connected to one another).

Figure 320 shows a Boundary Routing topology with a NETBuilder II system as the central node and three model 421 SuperStack II NETBuilder bridge/routers as peripheral nodes. Three 3Com-recommended TAs with BRIs connect to the NETBuilder II system with an HSS V.35 3-port module installed.

Figure 320 Boundary Routing Topology Using Multiple TAs with BRIs

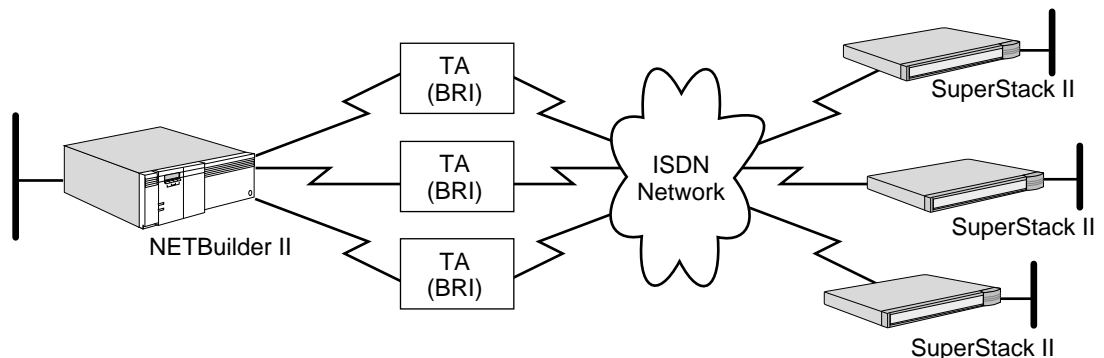
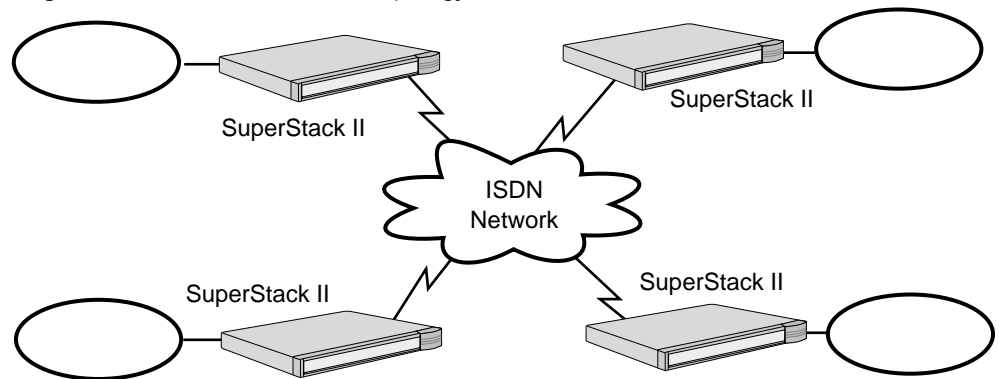


Figure 321 shows a traditional routed topology where four SuperStack II NETBuilder bridge/routers that represent small offices are connected to one another. In this topology, each small office communicates directly with one another.

Figure 321 Traditional Routed Topology (Meshed)

Although the ISDN topologies discussed in this section are the most common, many others can be used. For complete information on the 3Com-recommended TAs, see the documentation that accompanies these devices.

Deciding How to Use the ISDN Interface

Before cabling and configuring the ISDN interface, you need to decide how you want to configure the B and D channels.

The following are some basic questions that should help you decide:

- Do you want to use one B channel only, a combination of one B channel and one data terminal equipment (DTE) interface, or both B channels?
- How do you want to use each B channel? As an interface to a primary line or as an interface to a secondary or backup line?
- If you use a B channel as an interface to a primary, dial-up line, then do you want to manually dial and hang up or use dial-on-demand?

Dial-up is the method through which a call is placed through the D channel from one ISDN device to another ISDN device, and a 64 kbps circuit-switched B channel connection is established between the two devices. After a connection is established, data is transmitted over the B channel.

- If you use a B channel as a backup means of transmitting data, will the backup line connect the same site or a different site? Do you want the backup line to come up if the primary line fails (disaster recovery if backup to same site or network resiliency if backup to a different site) or if the primary line becomes congested and needs more bandwidth (bandwidth-on-demand)?
- Do you want to use static or dynamic paths?

Table 321-1 lists three common scenarios in which the ISDN interface is used. The figures that follow this table provide more information.

Table 321-1 Common Topologies Using ISDN Interface

| Type of Topology | Primary Interface | Primary Line Type | Secondary Interface | Secondary Line Type |
|--|---|-------------------|---------------------|---------------------|
| Boundary Routing topology using redundant routes for network resiliency/ISDN used as backup to Frame Relay or X.25 | DTE interface running X.25 or Frame Relay | Leased | ISDN interface | Dial-up |
| Boundary Routing topology using disaster recovery, bandwidth-on-demand, and network resiliency/ISDN as backup to same site or different site | DTE interface running PPP | Leased | ISDN interface | Dial-up |
| Traditional routed topology/ISDN as primary using dial-on-demand | ISDN interface | Dial-on-demand | None | None |

Bandwidth management is a process that applies static bandwidth, dynamic bandwidth, or a combination of these, to provide the ISDN and serial ports using PPP with the bandwidth they need to meet current requirements. Bandwidth management thinks in terms of *unrestricted*, available resources, or resources configured for a specific function such as disaster recovery only, instead of in terms of primary and secondary lines. Bandwidth management dynamically allocates or de-allocates available resources as necessary to manage link traffic. After reading the conceptual information, see the Configuring Port Bandwidth Management chapter for configuration steps.

Figure 322 shows a Boundary Routing topology where one NETBuilder II bridge/router is connected to a SuperStack II NETBuilder bridge/router through a DTE interface over which Frame Relay or X.25 is running, while the other NETBuilder II bridge/router is connected to the same SuperStack II NETBuilder bridge/router through an ISDN line over which PPP is running. The line running Frame Relay or X.25 is considered the primary line, while the ISDN line is considered the secondary line. In this topology, the secondary line is configured to come up only if the primary line fails, which provides a redundant route for network resiliency.

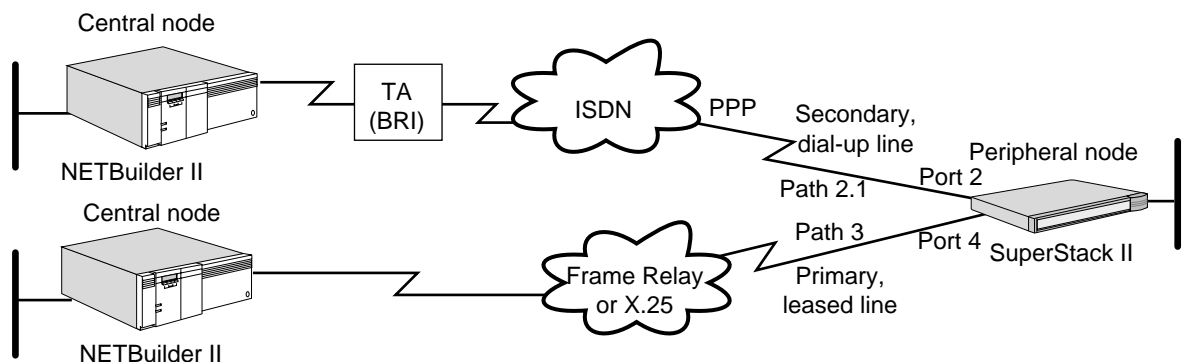
Figure 322 ISDN Used as Backup to Cloud Technology

Figure 323 shows additional Boundary Routing topologies. These topologies illustrate using an ISDN line as a backup to the same site or device, or as a backup to a different site or device.

In the first topology shown in this figure, two lines connect the NETBuilder II bridge/router to the SuperStack II NETBuilder bridge/router. The first line connects the two devices through a DTE interface over which PPP is running, while the second line connects the two devices through an ISDN path over which PPP is running. Both interfaces or paths are mapped to port 3. The DTE line is considered the primary line, and the ISDN line is considered the secondary line. In this topology, the secondary line is configured to come up only if the primary line fails (disaster recovery), or is overwhelmed by traffic and needs additional bandwidth (bandwidth-on-demand).

In the second topology shown in this figure, two lines connect two NETBuilder II bridge/routers to a SuperStack II NETBuilder bridge/router. The first line connects the two devices using a DTE interface running PPP, while the second line connects the two devices using an ISDN path running PPP. The DTE line is considered the primary line, while the ISDN line is considered the secondary line. In this topology, the secondary line is configured to come up only if the primary line fails (redundant route for network resiliency).

Figure 323 ISDN as Backup to Serial Line Running PPP (Same or Different Site)

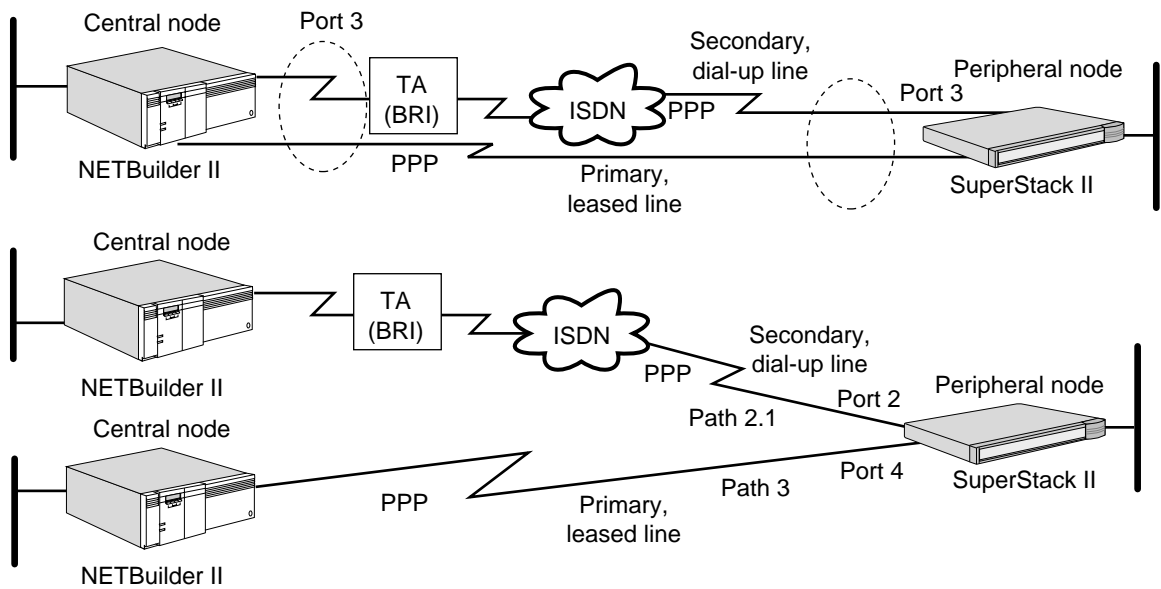
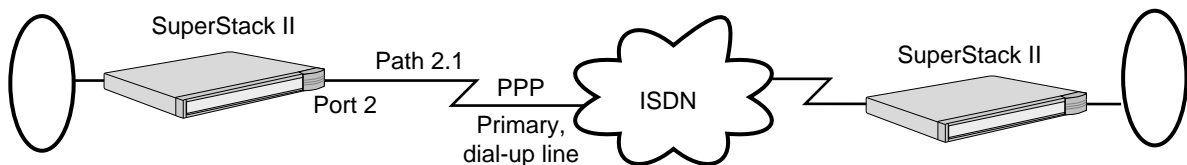


Figure 324 shows a traditional routed topology where a model 527 SuperStack II NETBuilder bridge/router is connected to another model 527 bridge/router through an ISDN path over which PPP is running. Because the ISDN line provides the only connection between these two devices, it is considered the primary line. In this topology, this line is configured to come up only when there is a demand for it (dial-on-demand).

Figure 324 ISDN as Primary Using Dial-on-Demand



Although the examples described in the preceding paragraphs are the most common, many other examples exist. For more information on Boundary Routing using network resiliency, disaster recovery, and bandwidth-on-demand, see the Configuring Boundary Routing System Architecture chapter. For information on dial-up, including more information on disaster recovery and bandwidth-on-demand, see the Configuring Port Bandwidth Management chapter.

Disabling Phantom Power

A Network Termination 1 (NT1) and a power supply are required for every ISDN line in North America. Your service provider or telephone company may provide you with an NT1 and power supply for a small monthly fee, or, you may want to purchase it from an ISDN equipment vendor. The NT1 and power supply may come in a single standalone box or the two may be in separate units. In this discussion, the two units together will be referred to as an NT1.

Two kinds of NT1s are currently in use in North America, differentiated by the data encoding scheme used in the transmission of data between the NT1 and the telephone company's equipment. The two data encoding schemes are called 2B1Q (two bits mapped into one quaternary symbol) and AMI (Alternate Mark Inversion). The 2B1Q scheme is the dominant method in use today. The AMI scheme is older and rarely used.

Two power sources are available from an NT1 for CPE equipment. An ISDN telephone uses one power source. The SuperStack II NETBuilder bridge/router does not use either one for power. Instead, it detects the presence or absence of phantom power and can determine whether or not a telephone cord is plugged in.

However, not all NT1s provide phantom power. The AMI NT1 from AT&T does not. If you are connecting the SuperStack II NETBuilder bridge/router to an NT1 that does not provide phantom power, you must turn off phantom power detection before you can dial successfully. To turn off phantom power detection, set the value of the `-PATH PhantomPower` parameter to `Disable`. For more information on this parameter, see the `PATH Service Parameters` chapter in *Reference for Enterprise OS Software*.



Phantom power is not supported on the HSS 8-port BRI modules.

Setting Up the Remote Device

After you have configured the ISDN device at the other end of the ISDN network (use the documentation that accompanies that device), no additional configuration is necessary for that device to interoperate with your SuperStack II NETBuilder bridge/router with an ISDN interface.

How the ISDN Interface Works

This section provides conceptual information on aspects of the ISDN interface that require further explanation.

Basic Rate Interface

SuperStack II NETBuilder bridge/routers with an ISDN interface provide ISDN connectivity through a BRI. This interface consists of two full-duplex B channels operating at 64 kbps and one full-duplex D channel operating at 16 kbps (2B + D). The two B channels transmit data, while the D channel is used for call processing with the ISDN switch.

Because the BRI consists of multiple B channels over which data can be transmitted, a path numbering convention has been devised. For complete information on this convention, see the *Configuring Basic Ports and Paths* chapter or the installation guide that you received with your SuperStack II NETBuilder bridge/router.

Paths 2.1 and 2.2 correspond to the B channels. However, a particular B channel is not statically bound to a particular path. At one time path 2.1 could use B channel 1, while at another time the same path could use B channel 2.

Some parameters that you must configure to set up the ISDN environment are connector-specific, that is, they are generically applicable to the ISDN operating environment, not to any one specific B channel. Connector-specific parameters require that you specify the connector number only. Other parameters are channel-specific, that is, they apply specified configurations to an individual channel and the physical connector which it is associated with.

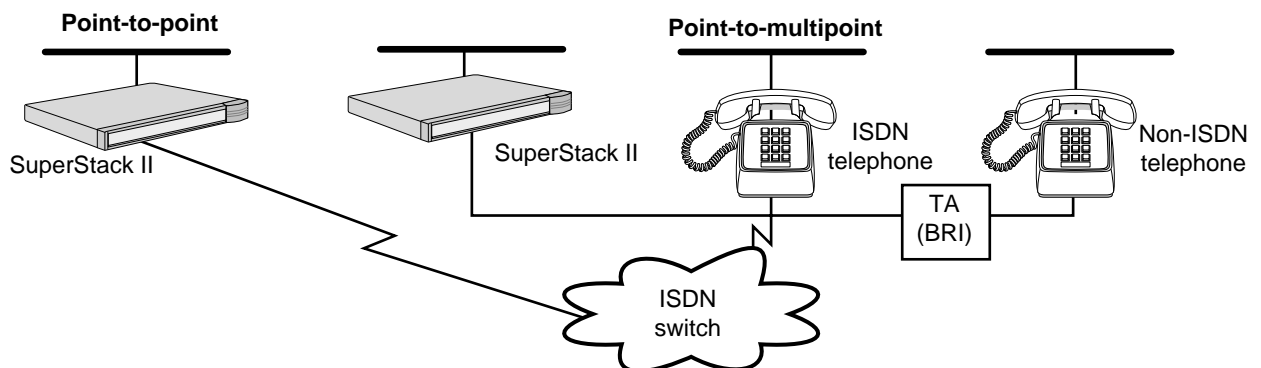
Channel-specific parameters require that you specify the connector number and the channel number. If you are unsure as to whether you need to specify the connector and channel numbers or just the connector number, see the description of that particular parameter in *Reference for Enterprise OS Software*.

Point-to-Point and Point-to-Multipoint Configurations

Your SuperStack II NETBuilder bridge/router with an ISDN interface can be a device in a point-to-point or point-to-multipoint configuration. A point-to-point configuration is a topology where a single device is connected to an ISDN line. A point-to-multipoint configuration is a topology where up to eight devices, including bridge/routers (SuperStack II NETBuilder bridge/router with an ISDN interface and other non-ISDN 3Com bridge/routers connected to a TA) and telephones (ISDN and non-ISDN connected to a TA), are connected to an ISDN line. The point-to-multipoint configuration is implemented using a passive S-bus.

Figure 325 shows two network topologies. The first topology (far left) is a point-to-point configuration; the second topology shows a point-to-multipoint configuration. Each configuration is connected to an ISDN switch through an ISDN line.

Figure 325 Point-to-Point and Point-to-Multipoint Configurations



How Incoming Calls Are Accepted

This section explains how SuperStack II NETBuilder bridge/routers with an ISDN interface decide to accept an incoming call from an ISDN switch.

SuperStack II NETBuilder bridge/routers with an ISDN interface use the following types of call compatibility criteria to determine whether or not to accept an incoming call from an ISDN switch:

- Bearer capability
- ISDN addressing

Some bearer capability criteria are fixed and cannot be changed, while others are determined by user configuration. The following sections describe each type of call compatibility criteria.

The ISDN specifications provide other compatibility criteria called low-layer compatibility and high-layer compatibility information elements that can be used to determine incoming call acceptability. SuperStack II NETBuilder bridge/routers with an ISDN interface do not use low-layer compatibility and high-layer compatibility information elements as criteria to determine whether or not to accept an incoming call from an ISDN interface.

Bearer Capability Compatibility

SuperStack II NETBuilder bridge/routers with an ISDN interface have the following fixed bearer capability criteria for an incoming call:

- It must be a 64K or 56K unrestricted digital data call. A voice call may be made to a telephone on a multipoint ISDN line to which the SuperStack II NETBuilder bridge/router is also connected. However, the SuperStack II NETBuilder bridge/router will not answer the call.
- It must be made in the circuit mode.

Calls that do not fulfill these criteria are rejected or ignored.

You can specify the rate at which data is to be transferred on a B channel that is to be connected by a call by using the `-PATH RateAdaption` parameter.

ISDN Addressing Compatibility

After an incoming call fulfills the bearer capability criteria, the following items must be determined:

- Which bridge/router or bridge/routers will answer the call?
- Which path will accept the call?

In a point-to-point configuration, where a single device is connected to an ISDN line, it is assumed that the bridge/router on the ISDN line will answer the call. Therefore, it is not necessary to configure the bridge/router as the device that will answer incoming calls.

In a point-to-multipoint configuration, where up to eight devices, including bridge/routers and ISDN and non-ISDN telephones, can be connected to an ISDN line, you must configure at least one bridge/router to answer the incoming calls.

After the incoming call is answered by at least one bridge/router in either the point-to-point and point-to-multipoint configurations, the bridge/router must also determine whether the call is to be connected to path 2.1 or 2.2. The ISDN switch selects a B channel over which to transmit a call and the bridge/router must determine the path that is to be connected with that B channel.

The selection of which bridge/router will answer a call and subsequently which path will accept a call is determined by how you address your ISDN paths. ISDN addresses consist of the following components:

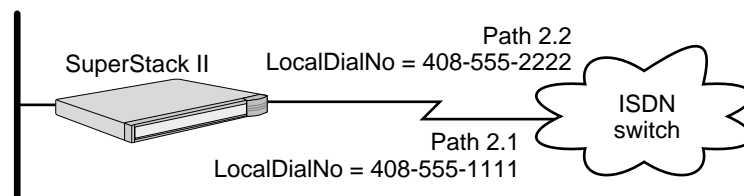
- Phone number
- Subaddress

For more information, see “ISDN Addressing” later in this chapter. You can assign a phone number and a subaddress to an ISDN path using the `-PATH LocalDialNo` and `-PATH LocalSubAddr` parameters, respectively.

Assign a phone number to an ISDN path under the following circumstances:

- You have a point-to-point or point-to-multipoint configuration and want to use static path and port binding. For example, you may want calls from the Boston office to be accepted by a particular path and port on a particular bridge/router, and calls from the Washington office to be accepted by another path and port on another bridge/router.
- You have a point-to-multipoint configuration and plan to use dynamic dial path pooling. For example, you may want calls to be accepted by any path in a dial path pool and then dynamically bound to a port. You should specify a phone number for at least one ISDN path on a bridge/router in this topology to ensure that at least one bridge/router will answer incoming calls.
- The telephone number you plan to specify for that path is unique among the other telephone numbers specified for ISDN paths in your point-to-point or point-to-multipoint configuration. Figure 326 shows a point-to-point configuration where both ISDN paths are assigned a unique phone number using the `-PATH LocalDialNo` parameter.

Figure 326 Assigning a Unique Phone Number to Multiple ISDN Paths

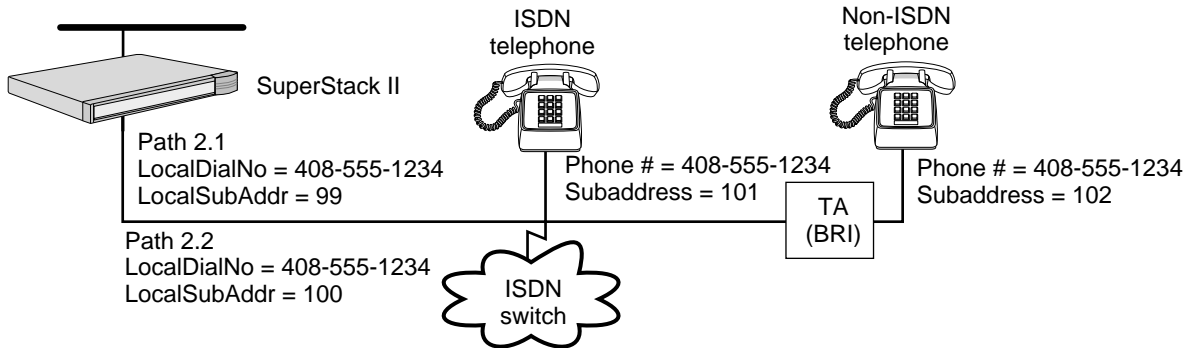


Suppose an incoming call specifying phone number 408-555-1111 arrives. As long as it is not engaged in another call, path 2.1 accepts the call based on the phone number specified in the incoming call. If path 2.1 was already engaged in another call or the phone number specified in the incoming call is different from that assigned to path 2.1, the call is rejected. If path 2.2 also cannot be used, the call is rejected by the bridge/router.

In addition to assigning a phone number, you should assign a subaddress to an ISDN path if the phone number is the same one that you plan to assign to another ISDN path in your point-to-point or point-to-multipoint configuration. Assigning the same phone number to all or some of the ISDN paths in a topology presents a problem: more than one ISDN path may attempt to accept a call. To resolve this problem, you can assign a subaddress to each of the bridge/router's ISDN paths with the same phone number using the `-PATH LocalSubAddr` parameter. For example, in the point-to-multipoint topology shown in Figure 327, four ISDN

paths have been assigned the phone number 408-555-1234. Unique subaddresses have also been assigned to the paths of each of these devices.

Figure 327 Assigning the Same Phone Number to Multiple ISDN Paths



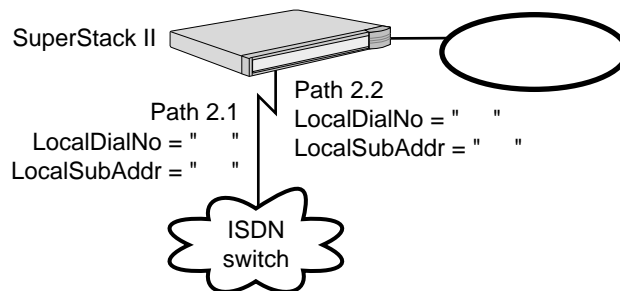
Suppose an incoming call specifying phone number 408-555-1234, subaddress 99, arrives. The SuperStack II NETBuilder bridge/router answers the call based on the phone number specified, and as long as it is not engaged in another call, path 2.1 accepts the call based on the subaddress specified. If path 2.1 was already engaged in another call or the phone number and subaddress specified in the incoming call is different from that assigned to path 2.1, the call is rejected. If path 2.2 also cannot be used, the call is rejected by the bridge/router.



Not all telecommunications carriers allow you to assign the same phone number to multiple paths. When you contact your carrier to acquire support services, verify that they support this feature. You must also specify that you will be using subaddresses.

Do not assign a phone number or a subaddress to an ISDN path if you have a point-to-point configuration and plan to use dynamic dial path pooling. Figure 328 shows a point-to-point configuration where a phone number and subaddress have not been assigned for both ISDN paths.

Figure 328 Point-to-Point Configuration Without Specified Phone Number and



Subaddress

Suppose an incoming call arrives that specifies a particular phone number and subaddress. The bridge/router ignores the ISDN addressing information provided by the incoming call and not use it as criteria to determine which path should accept the call. Either path can accept the call provided that they are not engaged in another call. Criteria at higher layers of the OSI Model will determine the port to which the path will be bound to transmit this particular call.

After you have assigned phone numbers to ISDN paths and an incoming call arrives, an algorithm attempts to match the incoming phone number with the phone number specified using the `-PATH LocalDialNo` parameter. This algorithm compares the numbers in sequence from the end of the numbers toward the beginning. The length of the incoming phone number can be shorter than the length of the phone number configured using the `-PATH LocalDialNo` parameter. For example, although you can specify an international phone number using elements such as a dial prefix, country code, area code, and phone number through the `-PATH LocalDial No` parameter, a phone number composed of only an area code and phone number will be considered a match as long as the phone number you specified and the phone number that is received through the incoming call are the same.

Specifying an international phone number using the `-PATH LocalDialNo` parameter allows you to accept all calls, including international and local. If you want to restrict incoming calls to local calls only, then specify at most an area code and local phone number using the `-PATH LocalDialNo` parameter.

An algorithm also attempts to match the incoming subaddress with the subaddress specified using the `-PATH LocalSubAddr` parameter. The characters for the subaddress in the incoming message must exactly match those specified in the `-PATH LocalSubAddr` parameter.

For more information on static and dynamic paths and dynamic dial path pools, see the *Configuring Port Bandwidth Management* chapter. For more information on the `-PATH LocalDialNo` and `-PATH LocalSubAddr` parameters, see the *PATH Service Parameters* chapter in *Reference for Enterprise OS Software*.

ISDN Addressing An ISDN address is a phone number provided by your telecommunications carrier. The address can consist of the following elements:

| | |
|---------------------|--|
| Dial prefix | Identifies an international dialing code used when calling from one country to another. |
| Country code | Identifies the destination country or geographic area and is from 1 to 3 digits long. |
| Area code | Identifies a particular ISDN network within the previously defined country or geographic area. |
| Local phone number | Identifies a subscriber's phone number in the previously defined area code. |
| Remote phone number | Identifies the destination phone number. |
| Subaddress | Identifies the destination device within the subscriber's passive bus topology. |

The telecommunications carrier does not provide a subaddress; you must create your own subaddress. For information on when to use a subaddress, see "ISDN Addressing Compatibility" earlier in this chapter.



Not all telecommunications carriers allow you to assign the same phone number to multiple paths. When you contact your carrier to acquire support services, verify that they support this feature. You must also specify that you will be using subaddresses.

When setting up certain parameters such as -PORT DialNoList, you may need to specify a dial or phone number string consisting of a phone number and, if applicable, a subaddress. If you specify a subaddress, you must separate the phone number and the subaddress with a semicolon (;). The phone number can consist of a maximum of 30 characters, while the subaddress can consist of a maximum of 20 characters.

When specifying a phone number, valid characters include the digits 0 through 9, an asterisk (*), and the pound sign (#). Because the software ignores all other characters to the left of the semicolon that separates the phone number and subaddress, you can also specify special characters such as parentheses and dashes to distinguish the different elements that compose a phone number, and text characters to embed descriptive text in the string.

When you specify a phone number, each character entered (whether the software considers it valid or invalid) counts toward the maximum allowable number of characters.

When you specify a subaddress, valid characters include all ASCII or IA5 characters.

The following string is an example of a dial or phone number string that includes a subaddress:

```
Los Angeles Office 1-213-555-1000;200
```

In this dial or phone number string, the phone number consists of long distance dial prefix 1 (assuming that the bridge/router being configured is located in Santa Clara), phone number 213-555-1000, and the subaddress 200. The descriptive text to the left of the semicolon indicates that the phone and subaddress numbers are for the Los Angeles office.

Austel Semi Permanent Circuit Support

An Austel Semi Permanent Circuit (ASPC) is an ISDN 64 KB B-channel circuit that uses the ISDN D-channel Signaling Protocol to initiate a call. The ASPC is unique in that the circuit is controlled like a dial line, but is tariffed and operated as a leased line.

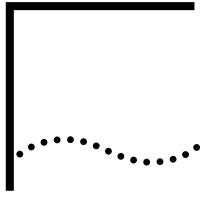
The ASPC line type designates that the interface supports the features of a leased line with a dial number. The ASPCNumber parameter allows you to associate a dial number with this class of ISDN leased line.

The basic rate ASPC relies on existing ISDN signalling to bring up a call. A special telephone number, the ASPC number, signals that an ASPC should be used for the call. Once connected, the line can be disconnected by the service provider at any time. The interface may deactivate or a DISCONNECT/RELEASE ACK may be received. If this situation occurs and the connection is again required, the originator redials the call until another connection is established.

Generally the port and path based dial parameters have no affect on an ASPC call. An ASPC call does not use the port and path parameters to support the call. The path is seen as a leased line.

When you change a BRI line (2.1 for example) to ASPC, the other BRI line on the same interface changes to ASPC unless it is a dialup line. With a dial line and an ASPC on the same interface, ASPC can be configured on B1 while B2 serves as a disaster recovery backup line.

For information about configuring ASPC on ISDN leased lines, see “Configuring Leased Lines” in the Configuring Port Bandwidth Management chapter.



CONFIGURING THE NETBUILDER II TO USE A WAN EXTENDER

This chapter describes how to configure the NETBuilder II bridge/router to use one or more WAN Extender systems to interconnect large numbers of remote LANs to a central site using channelized leased circuit services and switched circuit services.

This chapter also describes how to use the commands and parameters that are used for WAN Extender.



For conceptual information, see “How the WAN Extender Works” later in this chapter.

This chapter should be used in connection with the following guides:

- *WAN Extender 2T/2E Installation Guide*
- *WAN Extender 2T/2E Manager User's Guide*
- *Reference for Enterprise OS Software*

Circuit Services Supported

The WAN Extender provides virtual paths to be used by ports for leased and switched circuit services. The NETBuilder II bridge/router supports up to 75 virtual paths.

The following types of services are supported and can be used in configuring a NETBuilder II bridge/router to use a WAN Extender:

- Leased (permanent) circuit-based services:
 - Channelized T1
 - Channelized E1
- Switched circuit-based services:
 - Switched 56 (available with WAN Extender model 2T only)
 - ISDN Primary Rate Interface (PRI) (available with WAN Extender models 2T and 2E)

Configuring WAN Extender and NETBuilder II for Remote Connections

To enable a NETBuilder II bridge/router to use a WAN Extender to interconnect remote sites with a central site you need to configure the WAN Extender and the NETBuilder II bridge/router.

In your configuration, you have the following options:

- If you use an SCID (SysCallerID) to identify the remote callers for switched circuit services such as ISDN or Switched 56 or for T1 or E1 channelized leased lines, only 3Com NETBuilder bridge/routers at the remote sites can be interconnected with a WAN Extender to a NETBuilder II bridge/router at a central site.

- If you use Calling Line Identification Presentation (CLIP) to identify remote callers for ISDN dial-up paths, you can connect 3Com NETBuilder and other bridge/routers at the remote site to the NETBuilder II bridge/router at the central site.
- If a bridge/router port is being configured for ISDN dial-up paths, modem dial-up paths, and for a leased path, the port should be configured for SCID as well as CLIP to identify the port to the remote user. If a port is configured for SCID and CLIP, the CLIP configuration overrides the SCID configuration for incoming calls using ISDN dial-up paths. If the port is configured for something other than ISDN dial-up paths, SCID is used and CLIP is ignored.
- If you configure a T1 or E1 channelized leased lines with the -PORT WProfileList parameter, the remote bridge/routers do not need to supply an SCID string to identify themselves to the central NETBuilder II bridge/router. In this case, you can connect 3Com bridge/routers as well as other vendor's remote bridge/routers to a central NETBuilder II bridge/router.

You configure the WAN Extender according to the procedures in the *WAN Extender 2T/2E Manager User's Guide*. The configuration procedures in this chapter include only the WAN Extender configuration steps necessary to perform the NETBuilder II bridge/router configuration.

This chapter describes how to configure leased circuit lines. The sample configuration describes how to configure a channelized T1 leased line. If you are configuring E1 leased lines, enter E1 instead of T1 and WAN Extender 2E instead of WAN Extender 2T in the configuration procedure. This chapter also describes how to configure ISDN PRI switched-circuit lines, and briefly describes how to configure switched 56 lines, which is a similar configuration.

See "Remote Connection Configuration Considerations" later in this chapter for additional information to help you with the configuration procedures.

Requirements

To configure a NETBuilder II bridge/router to use a WAN Extender, the following hardware and software requirements must be met:

- WAN Extender software, any version up to 1.18, but 1.18 is recommended
- NETBuilder II software, version 9.1 or later
- An installed HSS RS-449 module
The HSS RS-449 *3-Port* module is not supported.
- A Dual Processor Engine (DPE) module, which support up to 512 virtual ports.

See *Installing the NETBuilder II Dual Processor Engine (DPE) Module* for more information.

Interconnecting Leased DS0s to Channelized T1

This configuration example shows how to achieve point-to-point interconnection of remote sites to a central site, using a leased DS0 (64 Kbps) circuit at each remote site and one or more channelized T1 (1536 Kbps) circuits at the central site.

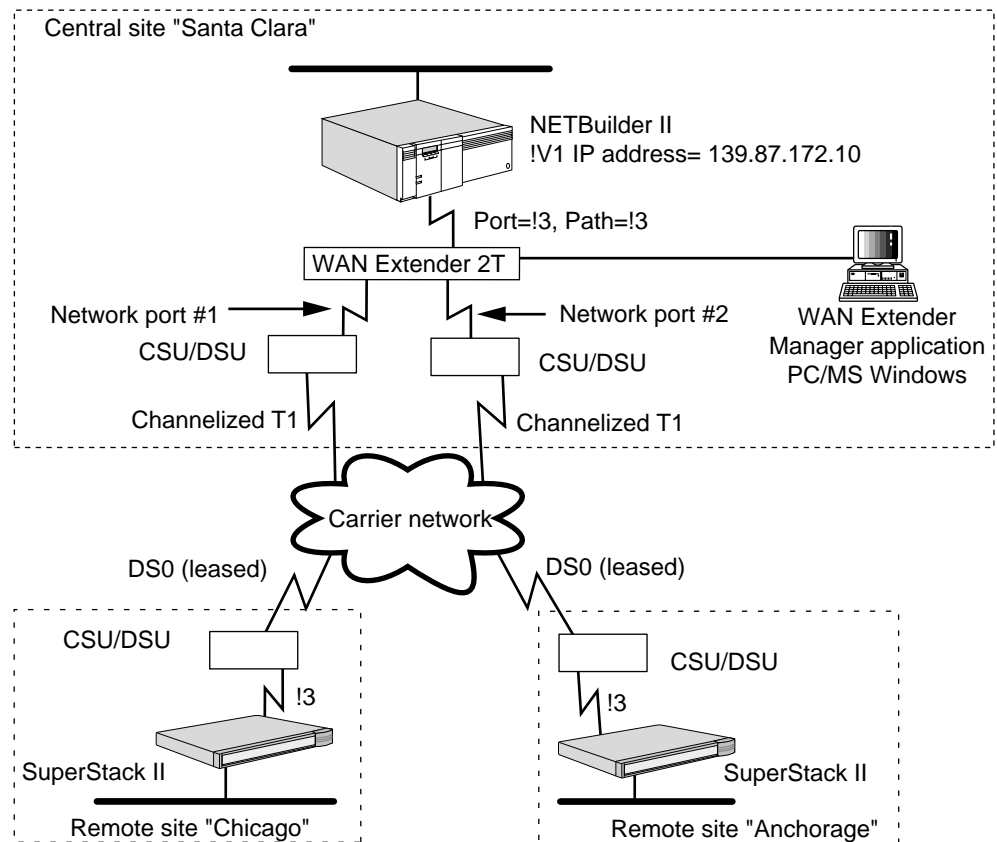
Although this example describes remote sites using single DS0s, remote sites can use a bundle of multiple DS0s. The WAN Extender can accommodate bundles of various numbers of DS0s running the channelized profile configuration, which results in a virtual path being automatically created on the NETBuilder II bridge/router at the central site with the appropriate baud rate for a bundle of

DS0s. For the bundles of DS0s to operate properly, a network services provider may be required to perform the proper mapping of DS0s through the network.

Figure 329 shows two channelized T1 circuits coming into a WAN Extender 2T, which is connected to a NETBuilder II bridge/router at the central site.

The bridge/routers at the remote sites are labelled as SuperStack II bridge/routers, although they could be any NETBuilder platform that supports a serial interface that can connect to a 64 Kbps (or larger) leased circuit with a suitable Channel Service Unit/Data Service Unit (CSU/DSU). NETBuilder II bridge/routers, SuperStack II NETBuilder bridge/routers, and OfficeConnect NETBuilder bridge/routers all qualify as a remote site.

Figure 329 Interconnecting Leased DS0s to Channelized T1



Configuring the WAN Extender

To configure the WAN Extender, use the *WAN Extender 2T/2E Manager User's Guide* and follow these steps with the WAN Extender Manager application:



For E1 configurations, substitute "E1" for "T1" and "WAN Extender 2E" for "WAN Extender 2T" in the steps that follow.

- 1 At the System Parameters window on the PC connected to your WAN Extender:
 - a Select WAN Extender 2T for System Type.
 - b Select Channelized T1 for Call Control for one or both network ports. If you are using only one port, it must be port number 1, and port number 2 must be set to Unused.

- c Set up the remaining parameters based on the type of existing network connections and according to the instructions in the *WAN Extender 2T/2E Manager User's Guide*.
- 2 From the Remote Site Profiles window, open and define a profile for each remote site to describe how the underlying connectivity to that site is achieved. In the Profile screen:
 - a Select Channelized T1 for Profile Type.
 - b Select the appropriate Outgoing Network Port.
 - c Click the check box(es) for the channel(s) that has been assigned for connectivity to the remote site.
 - d Complete your configuration by selecting a value for the Outgoing Call Circuit Type, deciding whether to enable or disable the Inverted HDLC on Selected Channels, and selecting values for the rest of the configuration items required for the profile creation as described in the *WAN Extender 2T/2E Manager User's Guide*.
- 3 Download the completed configuration file to the WAN Extender.
- 4 Reboot or reset the WAN Extender.

Configuring the NETBuilder II Bridge/Router

To configure the NETBuilder II bridge/router to use the WAN Extender, follow these steps using the terminal connected to the NETBuilder II bridge/router:

- 1 Set the owner of the NETBuilder II bridge/router port, which corresponds to the NETBuilder II bridge/router physical path to which the WAN Extender is connected, to WAN Extender.

For example, enter:

```
SETDefault !3 -PORT Owner = WanExtender
```

- 2 Set the baud rate on the WAN Extender-to-NETBuilder II bridge/router path to 4096 by entering:

```
SETDefault !3 -PATH Baud = 4096
```

The 4096 value is the only accepted baud rate value for the NETBuilder II bridge/router configuration to use the WAN Extender.

- 3 Set the clock for the physical path to External by entering:

```
SETDefault !3 -PATH CLock = External
```

- 4 Enable the physical path that corresponds to the serial connection by entering:

```
SETDefault !3 -PATH CONTROL = Enable
```

The WAN Extender and the NETBuilder II system will now synchronize using the 3Com proprietary WNI Protocol. After the synchronization, the NETBuilder II system path is in an UP state.

- 5 Create NETBuilder II bridge/router virtual ports, one for each remote site, to represent the logical attachment between the central site and the remote site by entering:

```
ADD !V1 -PORT VirtualPort SCID "Chicago"
ADD !V2 -PORT VirtualPort SCID "Anchorage"
```

The string within quotes uniquely identifies the remote site. This string must correspond to the configured -SYS SysCallerID parameter string of the remote site

NETBuilder II bridge/router. PPP packets received from the remote site with those strings will bind the ports to the appropriate virtual paths that represent the data channels through the WAN Extender.

- 6 If you do not want the remote bridge/routers that are attached to the channelized leased line to submit a SCID string to establish a connection with the centralized NETBuilder II bridge/router, enter:

```
ADD !V1 -PORT WProfileList "3 10"  
ADD !V1 -PORT WProfileList "3 11"
```

The RS-449 module, to which the WAN Extender is connected, is in slot 3 of the NETBuilder II bridge/router, and 10 and 11 are the leased WAN Extender profiles to which the virtual port is mapped. The profiles are mapped to one or more leased channels.

- 7 You must specify that the virtual port represents a permanent connection (as opposed to one that requires a dial-up) by entering:

```
SETDefault !V1 -PORT DialInitState = NoDialOut  
SETDefault !V2 -PORT DialInitState = NoDialOut
```

- 8 Specify the size of the largest packet that is transmitted or received between the NETBuilder II bridge/router and the SuperStack II bridge/router at the remote site.

If packets up to the maximum Ethernet size (1518) are to be bridged or routed across the connection, set the MaxRcvUnit parameter for a virtual port to 1518 by entering:

```
SETDefault !V1 -PPP MaxRcvUnit = 1518  
SETDefault !V2 -PPP MaxRcvUnit = 1518
```

If packets up to the maximum Fiber Distributed Data Interface (FDDI) or token ring size (4500) are to be bridged or routed across the connection, set the MaxRcvUnit parameter for a virtual port to the maximum value of 4500 by entering:

```
SETDefault !V1 -PPP MaxRcvUnit = 4500  
SETDefault !V2 -PPP MaxRcvUnit = 4500
```

For all NETBuilder II bridge/router virtual ports using WAN Extender virtual paths, the -PPP MaxRcvUnit parameter must be set to the appropriate maximum size value (4500 in this example). Failure to do so may result in the loss of network connections, and you may need to reboot the WAN Extender to recover connections.

When configuring the WAN Extender, the Maximum Data Buffer Size field must also be set properly to avoid losing network connections and to avoid rebooting the WAN Extender to recover connections. The Maximum Data Buffer Size field is configured on the System Parameters window of the WAN Extender Manager program. See the *WAN Extender 2T/2E Manager User's Guide* for configuration instructions.

If all the following three conditions exist, set the Maximum Data Buffer Size field to a value of 1750:

- Any WAN Extender virtual port on the NETBuilder II bridge/router is configured with PerPacket compression.
- The remote site is using a NETBuilder product.
- Only Ethernet packets are being compressed and bridged or routed through the WAN Extender (although the maximum Ethernet packet size is typically 1518 bytes, additional buffer space is required).

Configuring Other Protocols

After you have configured the NETBuilder II bridge/router to use the WAN Extender, you can configure bridging and other protocols (such as IP, IPX, and so on) on your NETBuilder II bridge/router for the virtual ports.

For example, if you are using IP over !V1, you can enter:

```
ADD !V1 -IP NETaddr = 139.87.172.10
SETDefault !V1 -IP CONTrol = ROute
SETDefault !V1 -RIPIP CONTrol = (Talk, Listen)
```

Verifying the Configuration

To verify that the WAN Extender and NETBuilder II bridge/router are operating as configured, follow these steps on the NETBuilder II bridge/router:

- 1 Check that the paths established through the WAN Extender are in the UP state by entering:

```
SHow -PATH CONFIguration
```



Because a WAN Extender virtual path does not bind to a port until a connection is established, some of the path parameters may not show on the Current Path Parameters display. Virtual paths used for leased channelized T1 or E1 are bound to a port when the NETBuilder II bridge/router and the WAN Extender synchronize with each other and the PPP negotiation is completed.

- 2 Check that all the ports that use WAN Extender virtual paths are in the UP state by entering:

```
SHow -PORT CONFIguration
```

- 3 Check the status of the PPP Protocol on each of the WAN Extender paths and ports by entering:

```
SHow -PPP STATUS
```

The Link Control Protocol (LCP) and all configured Network Control Protocols (NCPs) should be in the Open state under normal operation.

- 4 Display the configuration information for port 3 by entering:

```
SHow !3 -WE CONFIguration
```

See “Sample Configuration Verification Displays” later in this chapter for sample displays generated by this command. See the WE Service Parameters chapter in *Reference for Enterprise OS Software* for information about the -WE CONFIguration parameter.

- 5 Display the connection and data packet statistics between NETBuilder II bridge/router and the WAN Extender for all the ports by entering:

```
SHow -WE DevSTATistics
```

See “Sample Configuration Verification Displays” later in this chapter for sample displays generated by this command. See the WE Service Parameters chapter in *Reference for Enterprise OS Software* for information about the -WE DevSTATistics parameter.

- 6 Retrieve detailed information about incoming and outgoing calls for profile number 3 of the NETBuilder II bridge/router port 3, which is connected to the WAN Extender by entering:

```
SHow !3 -WE ProFIle 3 DETail
```


See “Sample Configuration Verification Displays” later in this chapter for sample displays generated by this command. See the WE Service Parameters chapter in *Reference for Enterprise OS Software* for information about the -WE ProFile parameter.

If you find problems with the configuration after verification, see “Troubleshooting” later in this chapter.

Interconnecting ISDN BRI Circuits to ISDN PRI

The configuration example shown in Figure 330 shows how to achieve point-to-point interconnection of two remote SuperStack II bridge/routers (through a WAN Extender) to a central site NETBuilder II bridge/router using ISDN Basic Rate Interface (BRI) circuits at the remote sites and ISDN PRI circuits at the central site. Two channels run through the same port at each of the remote sites. In this example, Northern Telecom DMS-100 is used as the ISDN switch-type service carrier.

The Multilink Protocol (MLP), which provides load balancing, is enabled in this example. Load balancing splits the data being sent over the available parallel PPP serial lines, and then properly sequences it at the receiving end. For more information about the Multilink Protocol, see the Configuring Boundary Routing System Architecture chapter.

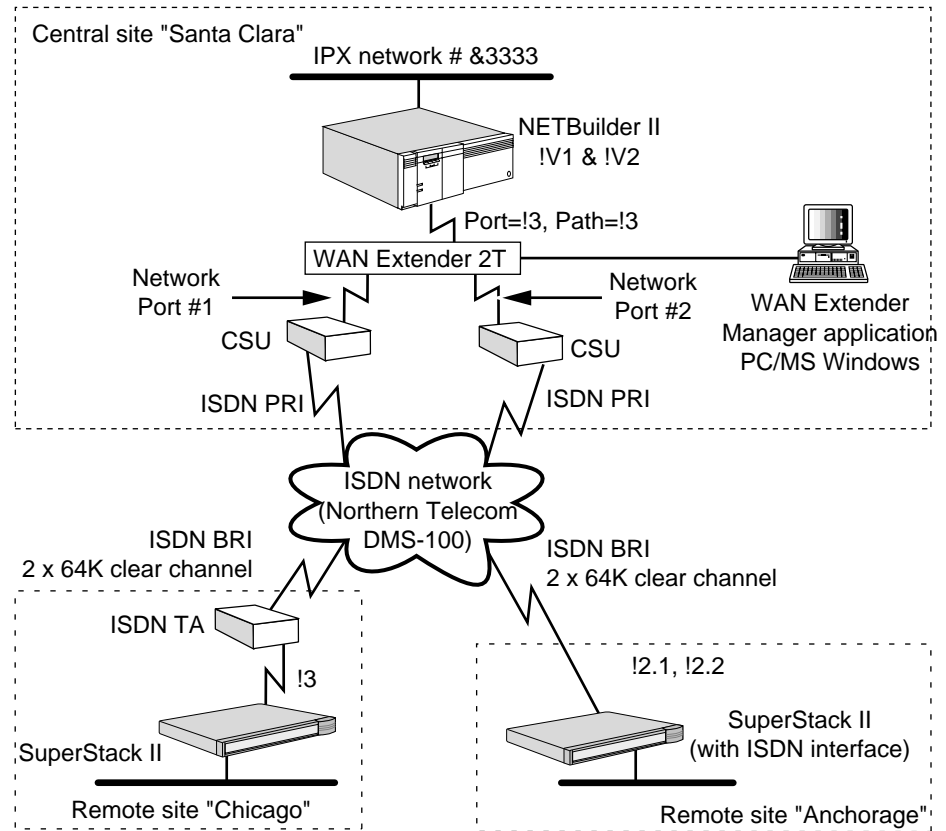
Although the routers at the remote site are labelled as SuperStack II, they can be any NETBuilder II bridge/router, SuperStack II NETBuilder bridge/router, or OfficeConnect NETBuilder bridge/router that supports:

- A serial interface that can connect to an ISDN BRI terminal adapter (TA).
- The ability to communicate with the TA for call establishment and teardown.
- Any NETBuilder platform with an integrated ISDN BRI TA.

Part of the following configuration instructions requires setting up the dial-up procedure, which includes selecting dial-up and remote-site identification options. See “Remote Connection Configuration Considerations” later in this chapter for information about these options. For more information about configuring and

using dial-up, see the Configuring the NETBuilder II to use a WAN Extender chapter.

Figure 330 Interconnecting ISDN BRI Circuits to ISDN PRI



Configuring the WAN Extender

To configure the WAN Extender, use the *WAN Extender 2T/2E Manager User's Guide* and follow these steps:

- 1 In the System Parameters window on the PC connected to your WAN Extender:
 - a Select WAN Extender 2T for System Type.
 - b Select ISDN for Call Control for both network ports.
 - c Set up the remaining parameters based on the type of existing network connections. See the *WAN Extender 2E/2T Manager User's Guide* for more information.
- 2 In the Port Parameters screen for each port:
 - a Select the switch specified by your network service provider from the Switch Type menu list.
 - b Select the switch variant specified by your network service provider from the Variant menu list.
 - c Select the call type specified by your network service provider from the Network Call Types Allowed field.
 - d Enable the B channels that are available for use by the WAN Extender in the Enabled Network B Channels check boxes.

- e Set up the remaining parameters based on the type of existing network connections and other configuration values entered. See the *WAN Extender 2E/2T Manager User's Guide* for more information.
- 3 In the Remote Site Profiles window, create a profile for each remote site to describe how the connection to that site is achieved.
In the Profile screen:
 - a Select ISDN for Profile Type.
 - b Select either Network Port 1 or Network Port 2 for Outgoing Call Network Port.
Both WAN Extender ports are connected to the same ISDN network in this example. Either port can be used to originate calls to any remote site or receive calls from any remote site.
If you are connected to two separate ISDN networks, you must assign each remote profile to a specific port.
 - c Select a circuit type supported by your ISDN configuration for Outgoing Call Circuit Type.
 - d Enter the remote site telephone number in the Outgoing Called Number field (optional).
 - e Enter the telephone number used for calling-party number matching when the WAN Extender receives an incoming call for Incoming Calling Number (optional).
 - f Select the proper Number Type and Numbering Plan.
 - g Set up the remaining parameters based on the type of existing network connections and other configuration values entered. See the *WAN Extender 2E/2T Manager User's Guide* for more information.
- 4 Make a note of the Profile ID that corresponds to each remote site.
This information is required to correctly configure the NETBuilder II bridge/router to establish the end-to-end connection. For this example, assume that the profile corresponding to the remote site "Chicago" was numbered 1 and 2. The profile corresponding to the remote site "Anchorage" was numbered 3 and 4.
- 5 Download the completed configuration file to the WAN Extender.
- 6 Reboot or reset the WAN Extender.
See the *WAN Extender 2T/2E Manager User's Guide* for more details on this procedure as well as other configuration options you may want to use.

Configuring the NETBuilder II Bridge/Router

To configure the NETBuilder II bridge/router, follow these steps:

- 1 Set the owner of the port that corresponds to the WAN Extender-to-NETBuilder II bridge/router connection to WAN Extender by entering:

```
SETDefault !3 -PORT Owner = WanExtender
```
- 2 Set the baud rate on the WAN Extender-to-NETBuilder II bridge/router connection to 4096 by entering:

```
SETDefault !3 -PATH BAud = 4096
```

The 4096 value is the only accepted value for the baud rate when configuring the NETBuilder II bridge/router to use the WAN Extender.

- 3 If you have configured the WAN Extender for caller identification through the PPP system identification data, you must allow the central site to identify itself to the remote sites by entering:

```
SETDefault -SYS SysCallerID = "Santa Clara"
```

- 4 Enable the path that corresponds to the serial connection by entering:

```
SETDefault !3 -PATH CONTROL = Enable
```

- 5 Create NETBuilder II bridge/router virtual ports, one for each remote site, to represent the logical attachment between the central site and the remote site by entering:

```
ADD !V1 -PORT VirtualPort SCID "Chicago"  
ADD !V2 -PORT VirtualPort SCID "Anchorage"
```

The string within quotes uniquely identifies the remote site. This string corresponds to the string configured on the remote router with the -SYS SysCallerID parameter. This identification is used during PPP link establishment to map the incoming call to the virtual port associated with the remote site.

- 6 If you want to use CLIP as the system to identify the remote sites to the central site, enter the ISDN phone number of each remote site into the CLIP database. For example, if the number for Chicago is 1-312-562-7758 with a subaddress 200 and Anchorage is 1-907-735-8758 with subaddress of 210, add them to the CLIP database by entering:

```
ADD !V1 -PORT CLIPList "1-312-562-7758;200"  
ADD !V2 -PORT CLIPList "1-907-735-8758;210"
```

The subaddress follows the ISDN number and a semicolon.

- 7 If you want to use CLIP, enable CLIP security checking of incoming calls to the specified port by entering:

```
SETDefault !3 -PORT DialRcvrState = AnswerCLI
```

After you have set the port for CLIP security, if the CLI dial string of the incoming call matches a dial string in the CLIPList database, an ISDN dial path is mapped to the port.

- 8 Specify the mapping between the virtual port that represents the logical attachment to the remote site and the WAN Extender profile that describes the underlying physical connection to the remote site by entering:

```
ADD !V1 -PORT DialNoList "3 1"  
ADD !V1 -PORT DialNoList "3 2"  
ADD !V2 -PORT DialNoList "3 3"  
ADD !V2 -PORT DialNoList "3 4"
```

In these commands, the first number within quotes is the number of the physical port to which the WAN Extender is connected. The second number is the profile ID on that WAN Extender that is used to make a call. V1 and V2 have two paths assigned to them. Two paths come up and become available when a call is put through the virtual port. The two available paths enable the Multilink Protocol (MLP) to use load balancing and split the data over the two paths if it becomes necessary.

The NETBuilder II bridge/router virtual paths go to available and UP states when the calls are completed, and if both sites are configured correctly.

- 9 Set the normal bandwidth to 128 kbps for virtual ports V1 and V2 by entering:

```
SETDefault !V1 -PORT NORMALBandwidth = 128
```

```
SETDefault !V2 -PORT NormalBandwidth = 128
```

This setting makes the Bandwidth Manager application bring up both paths for the virtual port to satisfy the port's normal bandwidth requirement of 128 kbps.

- 10 Enable MLP for V1 and V2 by entering:

```
SETDefault !V1 -PPP MlpControl = Enable
SETDefault !V2 -PPP MlpControl = Enable
SETDefault !V1 -PORT Control = Enable
SETDefault !V2 -PORT Control = Enable
```

- 11 Specify the dial-up initiation condition for each virtual port by entering:

```
SETDefault !V1 -PORT DialInitState = ManualDial
SETDefault !V2 -PORT DialInitState = ManualDial
```

- 12 Specify the size of the largest packet that will be transmitted or received between this NETBuilder II bridge/router and the SuperStack II bridge/router at the remote site.

If packets up to the maximum Ethernet size (1518) are to be bridged or routed across the connection, set the MaxRcvUnit parameter for a virtual port to 1518 by entering:

```
SETDefault !V1 -PPP MaxRcvUnit = 1518
SETDefault !V2 -PPP MaxRcvUnit = 1518
```

If packets up to the maximum FDDI or token ring size (4500) are to be bridged or routed across the connection, set the MaxRcvUnit parameter for a virtual port to the maximum value of 4500 by entering:

```
SETDefault !V1 -PPP MaxRcvUnit = 4500
SETDefault !V2 -PPP MaxRcvUnit = 4500
```

The -PPP MaxRcvUnit parameter must be set to the appropriate maximum size value (4500 in this example) for all NETBuilder II bridge/router virtual ports using WAN Extender virtual paths. Failure to do so may result in the loss of network connections, and you may need to reboot the WAN Extender to recover connections.

When configuring the WAN Extender, the WAN Extender Maximum Data Buffer Size field must also be set properly to avoid losing network connections and to avoid rebooting the WAN Extender to recover connections. The WAN Extender Maximum Data Buffer Size field is configured on the System Parameters window of the WAN Extender Manager program. See the *WAN Extender 2T/2E Manager User's Guide* for configuration instructions.

If all the following three conditions exist, set the Maximum Data Buffer Size field to a value of 1750:

- Any WAN Extender virtual port on the NETBuilder II bridge/router is configured with PerPacket compression
- The remote site is using a NETBuilder product
- Only Ethernet packets are being compressed and bridged or routed through the WAN Extender (although the maximum Ethernet packet size is typically 1518 bytes, additional buffer space is required)

Configuring Other Protocols

After you have configured the NETBuilder II bridge/router to use the WAN Extender, configure bridging and other protocols (such as IP, IPX, and so on) on your NETBuilder II bridge/router for the virtual ports.

For example, if you are using IPX over !V1, you enter:

```
SETDefault !V1 -IPX NETnumber = &123
SETDefault -IPX InternalNET = &3333
SETDefault !V1 -NRIP CONTrol = NoPEriodic
SETDefault !V1 -SAP CONTrol= NoPEriodic
SETDefault !V1 -IPX CONTrol = IpxWan
```

Verifying the Configuration

To verify that the configuration is correct, and that the WAN Extender and NETBuilder II bridge/router are operating as configured, follow these steps on the NETBuilder II bridge/router:

- 1 Try to establish a link with the remote site by entering:

```
Dial !V1
Dial !V2
```

- 2 Check that all paths established through the WAN Extender are in the UP state by entering:

```
SHow -PATH CONFIguration
```



Because a WAN Extender virtual path does not bind to a port until a connection is established, some of the path parameters may not show on the Current Path Parameters display. Virtual paths used for dial-up connections do not bind with a port until an outgoing call is completed or an incoming call is accepted.

- 3 Check that all virtual ports defined through the WAN Extender are in the UP state by entering:

```
SHow -PORT CONFIguration
```

- 4 Check the status of the PPP protocol on each of the ports established through the WAN Extender by entering (the LCP and all configured NCPs should be in the Open state):

```
SHow -PPP STATus
```

If you find problems with the configuration after verification, see “Troubleshooting” later in this chapter.

Configuring Switched 56 Circuits

The WAN Extender 2T can be configured to support remote sites that are connected through switched 56 circuits. Switched 56 circuits share many of the dial-up characteristics described in “Interconnecting ISDN BRI Circuits to ISDN PRI” earlier in this chapter.

The WAN Extender supports called ID and PPP system identification, but does not support incoming caller identification based on caller ID.

To configure the WAN Extender to support switched 56, follow these steps:

- 1 On the System Parameters window, select Switched 56 for call Control.
- 2 From the Remote Site Profiles window, select a profile for a remote site connected through a switched 56 circuit.
- 3 In the Profile window:
 - a Select Switched 56 for Profile Type.

- b Select Either (Port 1 preferred) or Either (Port 2 preferred) for Outgoing Call Network Port.
 - c Enter the remote site's telephone number in the Outgoing Called Number field.
- 4 Repeat steps 2 and 3 for each remote site connected with a switched 56 circuit.

Remote Connection Configuration Considerations

This section describes information you need to consider before configuring the NETBuilder II bridge/router to use ISDN or switch 56 switch-circuit services through the WAN Extender to link remote sites with a central site.

Dial-Up Options

ISDN circuits are switched circuits and require the execution of end-to-end call set-up procedures before a link between a remote site and the central site can be established. The Enterprise OS software offers several alternatives for determining when to establish the link. All of the options described here are available for and apply to links that are established through the WAN Extender. For more details about the dial-up alternatives, see the Configuring Port Bandwidth Management chapter.

Operator-Initiated Dialing (Manual Dial)

In operator-initiated dialing (manual dial) mode, you can use the Dial command from the NETBuilder II bridge/router to initiate the link between the remote site and the central site. The HangUp command terminates or tears down the link.

Scheduled Dial

Scheduled dial mode is a variation of the operator-initiated (manual) dialing. Instead of entering the Dial and HangUp commands, you can create NETBuilder macros that contain these commands, and you can use the SCHEDuler Service to specify when (day of week, time of day) these macros should be executed.

Auto Dial

Auto dial mode is another variation of the operator-initiated (manual) dialing. Instead of entering the Dial command to establish the link, the software automatically makes a call to the remote site at system initialization whenever the virtual port is enabled or whenever the WAN Extender-to-NETBuilder II bridge/router port or path is enabled.

Dial-on-Demand

In dial-on-demand mode, the Enterprise OS software automatically establishes the link when any user data needs to be forwarded between the remote and central sites, and disconnects the link when there is no outgoing traffic.

Because ISDN circuits can be established or disconnected on demand, configurations can be created where the number of remote sites is greater than the total number of ISDN B channels at the central site. This type of configuration is called *oversubscription*.

Oversubscription is useful for internetworking a large number of remote sites to a central site if the number of remote sites that need to be simultaneously connected to and communicating with the central site does not exceed the total number of available ISDN B channels at the central site.

The total number of remote sites cannot exceed the maximum number of central site virtual ports the Enterprise OS software can support. Each remote site attachment, whether active or not, must be represented as a unique virtual port at the central site.

Remote Site Identification Options

The WAN Extender or the NETBuilder II bridge/router must identify the originator of an incoming call so that the call can be mapped to the NETBuilder II bridge/router port associated with the remote site. For ISDN-based WAN Extender channels, several caller identification options are available. For more information about remote site identification options, see the *WAN Extender 2T/2E Manager User's Guide*.

ISDN Caller ID on the WAN Extender

Using ISDN Caller ID on the WAN Extender is the most efficient and cost-effective way to map incoming calls to NETBuilder II bridge/router ports. The call does not have to be completed (and therefore no charges incurred) before the call or the caller can be validated. The PPP system identification method requires that the call be completed before the caller can be validated.

In this method, the WAN Extender attempts to match the ISDN caller ID to the Incoming Calling Number fields in the ISDN remote site profiles, and it passes the incoming call (referencing the profile ID) to the NETBuilder II bridge/router.

The WAN Extender can be configured to reject any incoming call if no matching ISDN caller ID profile (or called ID profile) can be found on the WAN Extender. This capability is called *call filtering*.



Make sure incoming caller ID is supported across all network providers between sites before attempting this method of remote site identification.

ISDN Called ID on the WAN Extender

If multiple ISDN numbers (one for each remote site) are subscribed at the central site, for example, using a direct inward dialing (DID) numbering plan, the ISDN called ID caller identification method can be used. The WAN Extender attempts to match the ISDN called ID to the Incoming Called Number fields in the ISDN remote site profiles, and it passes the incoming call (referencing the profile ID) to the NETBuilder II bridge/router. This form of caller identification occurs after checking for an ISDN caller ID match.

PPP System ID Data on the NETBuilder II Bridge/Router

The PPP system ID data caller identification method is 3Com NETBuilder proprietary. If no ISDN caller ID-based or ISDN called ID-based mapping can be done, the WAN Extender relays the call to the NETBuilder II bridge/router. The NETBuilder II bridge/router accepts the call, establishes the link using PPP, and waits for PPP system identification data to arrive. This data is then used to associate a virtual port with the caller.

This method uses the unique string specified in the creation of a WAN Extender virtual port. The string that is received from the remote site is the value of the -SYS SysCallerID parameter of a remote NETBuilder bridge/router.

Call filtering must be disabled on the WAN Extender for this method to be used. When call filtering is disabled, the WAN Extender relays any calls from remote sites to the NETBuilder II bridge/router, whether a matching profile was found or not.

The NETBuilder systems at the remote sites must also use the PPP system identification data to identify the central site as the caller when they receive incoming calls.

CLIP Service Configuration

If you configure a NETBuilder bridge/router port for Calling Line Identification Presentation (CLIP) to identify remote callers for ISDN dial-up paths, you can connect NETBuilder and other vendors' bridge/routers to the NETBuilder II bridge/router at the central site.

To use CLIP, you must turn off call filtering when configuring the WAN Extender.

Customizing the Configurations

This section describes some WAN Extender configuration alternatives so that you can customize the configuration to your needs. These configurations are done through the Windows-based WAN Extender Manager application running on your PC connected to the WAN Extender console port.

ISDN H0 Support (WAN Extender 2T Only)

The WAN Extender 2T can be configured to establish H zero (H0) (384 kbps) ISDN PRI calls if you have purchased that capability from your ISDN service provider.

To configure for H0 calls, follow the steps described in "Interconnecting ISDN BRI Circuits to ISDN PRI" earlier in this chapter, and then follow these steps:

- 1 In the Port Parameters window for each network port capable of accepting or originating H0 calls, select 384Kbps for Network Call Types Allowed.
- 2 In the Remote Site Profiles window, select a profile that represents a remote site capable of accepting an H0 call.
- 3 In the Profile window, select 384Kbps for Outgoing Call Circuit Type.
- 4 Repeat steps 2 and 3 for each profile that represents a remote site capable of accepting an H0 call.
- 5 On the NETBuilder II bridge/router terminal console, set DialPathLimit to H0 for each WAN Extender port where H0 calls can be initiated or received.
- 6 Download the customized configuration to the WAN Extender, and then reset the WAN Extender.

Call Filtering

Call filtering limits caller identification to caller ID or called ID methods. The WAN Extender rejects incoming calls whose caller ID or called ID (based on the Incoming Calling Number field in the ISDN remote site profiles) does not match the WAN Extender profiles. This is the most efficient and cost-effective way to map incoming calls to NETBuilder II bridge/router virtual ports, because the call does not have to be completed (and no charges occur) before validating the call or the caller. The PPP system identification method requires the call to be completed first before the caller can be validated.

Make sure incoming caller ID is supported across all network providers between sites before attempting this method of remote site identification. See "ISDN Caller ID on the WAN Extender" earlier in this chapter for more information.

To enable call filtering on the WAN Extender for ISDN and switched 56 circuits, check the Call Filtering Enabled check box on the corresponding Port Parameter window.

Channel Bundling

The WAN Extender permits the bundling of channels or slots on a channelized T1 or E1 circuit to connect to a site that has a fractional T1 or E1 circuit provisioned, or is connected through a WAN Extender with a similar configuration.

To configure the WAN Extender for channel bundling, select more than one channel in the Channelized Profile window of WAN Extender Manager. See the *WAN Extender 2T/2E Manager User's Guide* for more information on this feature.

NETBuilder II Configuration Commands and Parameters

This section provides a brief description of the commands, Path Service parameters, and Port Service parameters that are used in the configuration of the NETBuilder II bridge/router for the WAN Extender.

For a description of all the WE Service parameters, see the WE Service Parameters chapter in *Reference for Enterprise OS Software*.

Commands

This section describes the command used in the configuration with a WAN Extender. For information on all commands, which you may use with the WAN Extender, see the Commands chapter in *Reference for Enterprise OS Software*.

DLTest

If the local link cable, local port, or NETBuilder II bridge/router serial interface adapter are not functioning correctly, the NETBuilder II bridge/router physical path connected to the WAN Extender will never transition to an UP state (visible by using the `SHow -PATH CONFIguration` command). By using the appropriate loopback connector on the NETBuilder II bridge/router serial interface adapter, and using the `DLTest` command as described in the Commands chapter in *Reference for Enterprise OS Software*, you can determine if the interface adapter is functioning correctly.

The WAN Extender can be placed into the loopback mode to use the `DLTest` command by placing the WAN Extender into loopback, by changing the baud rate from 4096 to 2048, and by changing the `-PATH Clock` parameter setting from External to TestMode. Placing the WAN Extender into the loopback mode allows you to use the `DLTest` command loopback test to verify the local link cable or WAN Extender limitations on passing packets.

PATH Service Parameters

This section describes the Path Service parameters used in the configuration with a WAN Extender. For a detailed description of the Path Service parameters, see the PATH Service Parameters chapter in *Reference for Enterprise OS Software*.

Baud

This parameter sets the correct baud rate for the local link between the NETBuilder II bridge/router and the WAN Extender. The baud must be set to 4096. This parameter does not apply to WAN Extender virtual paths because the baud rate for virtual paths is supplied to the NETBuilder II bridge/router by the WAN Extender.

CLock

This parameter determines how a bridge/router using serial interfaces derives its transmit clock. The WAN Extender provides clocking for the HSS RS-449 module

in the NETBuilder II bridge/router, so this parameter must be set to External. This parameter does not apply to WAN Extender virtual paths.

CONFiguration

This parameter displays the configuration and the current state of all paths, including WAN Extender-based virtual paths. The following is a sample display:

```
-----Current Path Parameters-----
Path  Name      Port  Ctrl  State  TlMode  Baud      Conn  Clock Line
      (kbps)
2     Path_2    2     Ena   Up     -       10000     -     -     -
3     Path_3    3     Ena   Up     -       4096      RS449 Ext  Leased
6     Path_6    6     Ena   Dwn   -       64        RS449 Ext  Leased
8     Path_8    8     Ena   Dwn   -       64        V35      Ext  Leased
V2    Path_V2   V2    Ena   Up     -       64        WE       Ext  Leased
V4    Path_V4   V4    Ena   Up     -       64        WE       Ext  Dialup
V5    Path_V5   -     Ena   Down  -       64        -        Ext  Dialup
```

The WAN Extender virtual path does not bind to a port until the connection is established. If the WAN Extender virtual path has not bound to a port, the Conn column on the Current Path Parameters display shows a hyphen instead of a value.

For WAN Extender virtual paths used as dial-up lines, a connection is established when an outgoing call is completed or when an incoming call is accepted. For channelized lines, the connection is established when the NETBuilder II bridge/router synchronizes with the WAN Extender and PPP negotiation is completed.

CONNector

This parameter specifies the connector type for a serial interface. When you change this parameter setting, you need to re-enable the corresponding path for the new parameter value to take effect for the path to which the WAN Extender is connected. This parameter must be set to RS449. This parameter does not apply to WAN Extender virtual paths.

CONTRol

This parameter enables or disables a path on the bridge/router. By disabling and enabling the path, all the values associated with the CONTRol parameter take effect. The only options that apply to WAN Extender-based virtual paths, or to the path the WAN Extender is connected to, are Enable and Disable. For the path the WAN Extender is connected to, Disable causes all virtual paths established through that WAN Extender to go down and become unavailable.

Enable causes the WAN Extender and the NETBuilder II bridge/router to go through a resynchronization procedure on the local link. The NETBuilder II bridge/router first attempts to retrieve WAN Extender system information and global and network port level configuration settings.

DialCONTRol

This parameter is a bit-mapped control parameter, which sets the path attributes for the dial-up paths. When configuring a NETBuilder II bridge/router for a WAN Extender, the WAN Extender virtual paths available for dial-up paths are set

automatically to the default values for the DialCONTRol parameter, except that the virtual paths are automatically set to DYNamic and not STATic.

DialPool

This parameter displays the dial pool status and configuration. This display shows all paths in the dial pool, all dynamic paths, both physical and virtual, the last time the path was used, the time when the current path became active, the external device type, and which ports have reserved the dial paths through the -PORT PathPreference parameter.

Because WAN Extender virtual paths do not bind to a port until a connection is established, virtual dial paths will not be reserved for specific ports through the -PORT PathPreference parameter. When you enter the SHow -PATH DialPool command, the virtual paths provided by WAN Extender to the dial-up pool are displayed, but the reservation of WAN Extender virtual paths to a particular port are not displayed.

For WAN Extender virtual paths used as dial-up lines, a connection is established when an outgoing call is completed or when an incoming call is accepted.

ExDevType

This parameter specifies and displays the external device type attached to a DTE connector. The HSS modules installed in a NETBuilder II bridge/router have an RS-232 or RS-449 DTE connector type. This parameter is used only with the dial-up path selection algorithm for matching destination phone numbers with dynamic dial ports. For NETBuilder II bridge/routers with a WAN Extender, this parameter is set automatically to WE or WEHO. This setting can be viewed, but not changed with the ExDevType parameter.

LineType

This parameter sets the type of line being used on a wide area interface. The options are Leased or Dial-up. For the physical path to which the WAN Extender is connected, this parameter must be set to Leased. The LineType for virtual paths is set automatically by the WAN Extender device driver to Dial-up for a dial-up channel, such as an ISDN or Switched 56 channel, and to Leased for a channelized virtual path. The LineType settings for virtual paths can be viewed but not changed with this parameter.

PORT Service Parameters

This section describes the Port Service parameters used in the configuration with the WAN Extender. For a detailed description of the Port Service parameters, see the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

CLList

This parameter adds (or deletes) a "dial string" (usually the ISDN phone number and subaddress) to a list of numbers to be used by the called party to map the incoming call to the appropriate port and to bind an ISDN dynamic path to the port to complete the call. This parameter is also used to screen out any calls that do not have a match in the CLList database.

The CLList entries can take effect only if -PORT DialRcvrState has been set to AnswerCLI. See the -PORT DialRcvrState parameter for more details.

The binding of a path to a port with a CLI number supersedes and ignores the binding between path and port set up by a system caller ID (SCID) number.

COMPResType

This parameter determines the compression type for virtual ports. The only type of compression available for virtual ports that are based on the WAN Extender is per-packet compression. The per-packet link-level option looks for repetitive patterns within a packet and replaces them with shorter length codes.

For more details on this parameter, see the Configuring Data Compression chapter in this guide and the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

CONFiguration

This parameter displays the configuration associated with WAN Extender-based virtual ports. For these ports, the Owner column contains PPP and the Paths column contains the SCID "SysCallerID" that was entered when the virtual port was added. For the NETBuilder II bridge/router physical port to which the WAN Extender is connected, the Owner column contains WE (WAN Extender), and the Paths column contains the number of the path to which the WAN Extender is connected.

```
-----Current Port
Parameters-----
Port      Name      Ctrl   State  Owner  Paths
1         Port_1   Ena    Up     Eth    1
2         Port_2   Ena    Up     Eth    2
3         Port_3   Ena    Dwn    Eth    3
4         Port_4   Ena    Dwn    WE     4
V2        Port_V2  Ena    Up     PPP    v1
          SCID"SanDiego"
V4        Port_V4  Ena    Up     PPP    v2
          SCID"SanJose"
V5        Port_V5  Ena    Down   PPP    -
```

DialNoList

This parameter adds, deletes, edits, and displays a list of phone numbers with their associated attributes (baud rate, phone number, and position in the list). The following is the syntax for this parameter:

```
ADD !<port> -PORT DialNoList "<phone no>" [Baud = <rate> (1.2-16000)]
  [Type = Modem | Bri | Sw56 | WE | WEH0]
  [Pos = <number>]
DELeTe !<port> -PORT DialNoList "<phone no>"
SHoW [!<port> | !*] -PORT DialNoList
```

If you specify WE or WEH0 as the Type value, the value entered for "<phone no>" is the NETBuilder II system port number the WAN Extender is connected to and a WAN Extender remote site's profile ID. (The remote site profile has the remote site phone number.)



For WAN Extender, the Baud rate specification is ignored. The baud rate associated with a virtual port is derived from the actual connection bandwidth.

DialRcvrState

This parameter determines whether a port answers calls or not. Set the DialRcvrState parameter to AnswerCLI if you want the bridge/router to try to match an incoming call to the port specified and to bind an ISDN dynamic path to the port. The binding only occurs if the ISDN number (and subaddress) of the incoming call matches the ISDN number in the CLList database. See the -PORT CLList parameter for more details. AnswerCLI applies only to incoming ISDN dial-up paths only. It has no effect on other types of dial-up paths.

DialStatus

This parameter displays a WAN Extender virtual port's path number, B channel number, and the network port if the path is up. The other fields in the display are the same as for other ports.

OWNer

This parameter indicates which NETBuilder II bridge/router physical ports are connected to WAN Extenders. All ports that use WAN Extender virtual paths use PPP as the data link protocol. This parameter does not indicate the data link protocol for a given port.

You do not need to configure any services on a port set to OWNer = WanExtender. The physical port and associated physical path are only used to support the WAN Extender virtual paths used by other ports in the system.

For example, to indicate that a WAN Extender is connected to the associated path for port 3, enter:

```
SETDefault !3 -PORT OWNer = WanExtender
```

PAtHs

This parameter assigns a path or multiple paths to the specified port, or assigns dial pool path resources to the specified port. Default ports must have a SCID string associated with them to allow SCID-based mapping of WAN Extender virtual paths to the ports over which incoming calls will arrive. For a complete description of this parameter, see the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

PathPreference

This parameter is not used for WAN Extender.

VirtualPort

This parameter creates a virtual port that represents a logical attachment with the WAN Extender to a network at a remote site. The virtual port specification includes a system identifier string, which may be used during incoming call setup time to associate the caller (a 3Com NETBuilder bridge/router at a remote site) with the corresponding virtual port. For example:

```
ADD !V1 -PORT VirtualPort SCID "Chicago"
```

Sample Configuration Verification Displays

This section provides sample displays and descriptions of the display elements for the SHow command and various WE Service parameters that are used to verify the

configuration. For a detailed description and syntax of all WE Service parameters, see the WE Service Parameters chapter in *Reference for Enterprise OS Software*.

Configuration Setting Displays

To display the current WAN Extender system and network configuration settings for the port entered as well as for its Local Management Interface (LMI) parameters and their settings, use the configuration parameter. If no port is specified, then the configuration information for all WAN Extender ports (and their owners) are displayed in ascending order.

For example, enter:

```
SHoW -WE CoNFIguration
```

A display similar to the following appears:

```
WAN Extender Configuration
System Port Parameters
KeepAliveInt                10
FullStatusFreq              6
ErrorThreshold              3
DialPathLimit               10, 2
WAN Extender Device Parameters
Version                     WANExtender Rel 10.1 T1   7/97
Type                       WAN Extender 2T
Name                       NB2_4
Max Data Pkt Size          1518 bytes
Network Port 0
switch type                 AT&T 5ESS
service variant             AT&T Custom
rate adapted                Disabled
call types allowed          56KB, 64KB_Clear
Network Port 1
enabled channels            0x00FFFFFF
circuit id                  Network Port 2
```

Connection and Data Packet Statistics Displays

You can display the connection and data packet statistics accumulated between the NETBuilder II bridge/router port and the WAN Extender that is connected to the port, and the statistics generated for the WAN Extender network ports by using the DevSTATistics parameter.

For example, enter:

```
SHoW !4 -WE DevSTATistics
```

A display similar to the following appears:

```
Statistics from WAN Extender out port !4:
Network Port 1:
  Profile misses                2
  Calls made                    24
  Calls received                2
  Out calls blocked             0
  Outgoing call bad profile     0
  Internal errors               0
Network Port 2:
  Channelized profile errors    0
  Internal errors               0
Packet Transfer Statistics:
  Packets out network ports     35167
  Pkts from NETBuilder dropped due to full queue 0
```

```

Packets received from network ports          34601
Pkts from network ports dropped due to full queue  0
LMI packets received from NETBuilder         9842
LMI Packets sent to NETBuilder              9923
Invalid DLCI occurrences from NETBuilder     9
Idle DLCI occurrences from NETBuilder       2

```

This command can only be entered as a UI command at the local console. This parameter is not available through Scheduler or Remote commands.

Incoming and Outgoing Calls Displays

You can retrieve information from the WAN Extender that is connected to the NETBuilder II bridge/router port for the incoming and outgoing calls made through the port using the ProFile parameter.

For example, enter:

```
SHow !4 -WE ProFile 4
```

A default summary display similar to the following appears:

```

----- Profile #4 from WAN Extender out port 4 -----
Outgoing called number..... 4962134
Outgoing calling number..... 9868404
Incoming called number..... 4962134
Incoming calling number.....

```

If you enter:

```
SHow !4 -WE ProFile 4 Detail
```

A display similar to the following appears:

```

----- Profile #4 from WAN Extender out port 4 -----
Calls made..... 7
In called number matches..... 0
In calling number matches..... 0
Version..... 0xA005
Description..... 9868404 -> 4962134
Profile type..... ISDN (0)
Network port..... 1
Outgoing called number type..... Subscriber
Outgoing called number plan..... ISDN
Outgoing called number..... 4962134
Outgoing calling number type..... Subscriber
Outgoing calling number plan..... ISDN
Outgoing calling number..... 9868404
Incoming called number..... 4962134
Incoming calling number.....

```

If you enter:

```
SHow !4 -WE ProFile 4 STATistics
```

A display similar to the following appears:

```

----- Profile #4 from WAN Extender out port 4 -----
Calls made..... 7
In called number matches..... 0
In calling number matches..... 0

```

Packet Counts Displays

You can display the WAN Extender-to-NETBuilder II bridge/router connection statistics as counted on each NETBuilder II bridge/router port with WAN Extender

set as Owner by using the -SYS Service STATistics parameter. The connection statistics displayed include packet counts for WAN Extender virtual paths and WAN Extender local-link operation statistics.

For example, enter:

SHow -SYS STATistics -WanExtender

See the Statistics Displays appendix for a sample display that comes up with this command and a description of the display elements.

Troubleshooting

If you have verified your configuration and have found problems with your system, you must troubleshoot the problems. This section describes what to check for in channelized leased-line configurations and switch-circuit configurations.

This section also provides information about WAN Extender and NETBuilder II bridge/router troubleshooting commands you can use to further verify and troubleshoot a configuration.

Troubleshooting Channelized Leased Configurations

To troubleshoot leased line problems, check to see if one or more of the following situations has occurred:

- The remote site is down or not connected.
- No profile is configured for the remote site, or if a profile is configured, it is configured incorrectly.
- You have configured your channelized leased line with the -PORT WEProfileList parameter so that you do not need to identify a remote site with a SCID identifier, and the profile ID configured for a virtual port was already in the database for another port. Or you have entered more than the maximum of 16 profileIDs for a given virtual port.
- You have not configured your channelized leased line with the -PORT WEProfileList parameter, and the SysCallerID (SCID) string set for the virtual port designated for the remote site does not match the Service SysCallerID parameter string of the remote site.
- The local port cable connecting to network is not connected properly or is faulty.
- The network port cabling is not connected properly or is faulty.

Troubleshooting Switch Circuit Configurations

To troubleshoot an ISDN or switch 56 switch circuit problems, check to see if one or more of the following situations has occurred:

- The remote site is down or not connected.
- No profile is configured for the remote site, or if a profile is configured, it is configured incorrectly.
- The SysCallerID (SCID) string set for the virtual port designated for the remote site does not match the Service SysCallerID parameter string of the remote site system.
- The local port cable connecting to network is not connected properly or is faulty.
- The network port cabling is not connected properly or is faulty.

Using WAN Extender Troubleshooting Commands

The WAN Extender console port provides access to a set of commands for verifying the WAN Extender configuration and for troubleshooting the system operation. These commands can be used to:

- Display the contents of the configuration file the WAN Extender is currently running, including system parameters, port parameters, and remote site profiles.
- Trace messages passing between the WAN Extender and the NETBuilder II bridge/router.
- Trace call control messages on ISDN links.
- Display statistics on profile usage, packet transfers, and incoming/outgoing call completions.
- Retrieve diagnostic information following a WAN Extender failure.
- Reboot the WAN Extender.



CAUTION: *The WAN Extender troubleshooting commands perform actions that may seriously impact the ability of the WAN Extender to accept and initiate calls. These commands should only be used under the close supervision of qualified 3Com support technicians and only during periods of light or no-call traffic.*

Accessing the WAN Extender Console Interface

Connect a PC to the console port on the WAN Extender rear panel using the same console link cable as used for the WAN Extender Manager application. On the PC, run a terminal emulation program that is configured as follows:

- 9600 baud
- 8 bits
- No parity
- 1 stop bit
- No software flow control (XON and XOFF are ignored)



If you are using the same PC as the WAN Extender Manager, make sure you have exited from that program before using a terminal emulation program. Failure to do so will result in an error message that the port is in use.

When the terminal emulation program is running, press the Enter key several times until an > prompt appears, indicating that the WAN Extender is ready to accept troubleshooting commands.

Command Descriptions

You can display the list of troubleshooting commands by entering:

WE??

The WAN Extender troubleshooting commands are case-sensitive. The following list describes these commands:

| | |
|------|---|
| WEvs | Displays system status. |
| WEvc | Displays connection states. |
| WEss | Displays system and network port-related configuration parameters, including version numbers. |

| | |
|--------------------|---|
| WEsp <profile-id> | Displays contents of the remote site profile specified in <profile-id> and the statistics associated with that profile. |
| WEpc | Displays packet count statistics for the local port and network ports. |
| WErb | Reboots the attached WAN Extender. |
| WElb | Sets the attached WAN Extender in local loopback mode. In this mode, all frames transmitted by the NETBuilder II system on the local port are looped back to the NETBuilder II system without any change. |
| WEdw | Disables watchdog timer. |
| WEtp | Do not use—for testing only. Toggles profiles mode (O = loaded, I = static). |
| WEts | Do not use—for testing only. Toggles system parameter mode (O = loaded, I = static). |
| WErx | Do not use—for testing only (use WEst instead). Toggles RX tracing (O = Off, I = On). |
| WEtx | Do not use—for testing only (use WEst instead). Toggles TX tracing (O = Off, I = On). |
| WEes | Do not use—for testing only. Edits on-board system profile. |
| WEcr | Dumps 68ec030 registers after a board-level panic. |
| WEsa | Do not use—for testing only. Sets system to stand-alone mode to make test calls. |
| WEca <profile-id> | Do not use—for testing only. Makes a test call in stand-alone mode using a remote site profile specified in <profile-id>. |
| WEhu <dcli> | Do not use—for testing only. Hangs up a test call made in stand-alone mode. <dcli> is the connection identifier associated with that call. |
| WEtl | Do not use—for testing only. TSI loopback for ACCUNET testing (no host). |
| WEst <trace-level> | Turns on or turns off tracing. A variety of trace options are available, each expressed as a hexadecimal value. |



Use of the WEst<trace-level> command can seriously affect the performance of the WAN Extender. Do not turn on large combinations of trace types during periods of high-call traffic on the WAN Extender.

To get a combination of trace types, add up the hexadecimal values corresponding to the individual types. For example, if you want to enable these three trace types:

- Trace SMI messages exchanged between the NETBuilder II bridge/router and the WAN Extender (0x0020)
- Trace the handling of calls (connection establishment and teardown) (0x0004)
- Trace unexpected execution paths (0x0001)

Enter:

Superuser WAN Extender TraceLevel 25

Table 81 lists the traces and their hexadecimal values.

Table 81 Traces and Their Hexadecimal Values

| Value | Meaning |
|--------|----------------------------------|
| 0x0000 | Turns trace off. |
| 0x0001 | Traces all LMI messages. |
| 0x0004 | Traces race conditions. |
| 0x0008 | Traces minor debug information. |
| 0x0010 | Traces the processing flow. |
| 0x0020 | Traces timer processing. |
| 0x0040 | Traces restart state machine. |
| 0x0080 | Traces span state machine. |
| 0x0100 | Traces error conditions. |
| 0x0200 | Traces control messages. |
| 0x0400 | Traces call control error paths. |
| 0x0800 | Traces call control flow. |

In addition to commands, the WAN Extender also supports ISDN link-level tracing on the network ports:

- To get basic level of tracing, at the WAN Extender console prompt, enter lowercase L and digit one. For example:

```
l 1
```

- To get expanded tracing, at the WAN Extender console prompt, enter lowercase L and digit two. For example:

```
l 2
```

- To turn off link-level tracing, at the WAN Extender console prompt, enter lowercase L and digit zero. For example:

```
l 0
```

The trace displays ISDN Layer 2 and Layer 3 call control messages exchanged between the WAN Extender and the network as shown in a display similar to the following:

| Ch# | Time | Direct | SAPI | TEI | C/R | Type | N(s) | N(r) | P/F | Size |
|-----|------|--------|------|-----|-----|--------|------|------|-----|------|
| 00 | 1AAF | Xmit | 00 | 00 | 0 | SABME | | | 1 | 0003 |
| 00 | 1AB3 | Rcvd | 00 | 00 | 1 | SABME | | | 1 | 0003 |
| 00 | 1AB3 | Xmit | 00 | 00 | 1 | UA | | | 1 | 0003 |
| 01 | 1AB4 | Rcvd | 00 | 00 | 1 | UA | | | 1 | 0003 |
| 00 | 1B7A | Xmit | 00 | 00 | 0 | Setup | 00 | 00 | 0 | 0027 |
| 01 | 1B7C | Rcvd | 00 | 00 | 0 | Setup | 00 | 00 | 0 | 0027 |
| 01 | 1B7C | Xmit | 00 | 00 | 1 | Prdng | 00 | 01 | 0 | 000E |
| 00 | 1B7D | Rcvd | 00 | 00 | 1 | Prdng | 00 | 01 | 0 | 000E |
| 00 | 1B7D | Xmit | 00 | 00 | 1 | RR | | 01 | 0 | 0004 |
| 01 | 1B7D | Xmit | 00 | 00 | 1 | Alrtng | 01 | 01 | 0 | 0009 |
| 01 | 1B7E | Rcvd | 00 | 00 | 1 | RR | | 01 | 0 | 0004 |
| 00 | 1B7E | Rcvd | 00 | 00 | 1 | Alrtng | 01 | 01 | 0 | 0009 |
| 00 | 1B7E | Xmit | 00 | 00 | 1 | RR | | 02 | 0 | 0004 |
| 01 | 1B7E | Xmit | 00 | 00 | 1 | Connct | 02 | 01 | 0 | 0009 |
| 01 | 1B7F | Rcvd | 00 | 00 | 1 | RR | | 02 | 0 | 0004 |
| 00 | 1B7F | Rcvd | 00 | 00 | 1 | Connct | 02 | 01 | 0 | 0009 |
| 00 | 1B7F | Xmit | 00 | 00 | 0 | ConAck | 01 | 03 | 0 | 0009 |

01 1B81 Rcvd 00 00 0 ConAck 01 03 0 0009

Using NETBuilder II Troubleshooting Commands

NETBuilder II bridge/router troubleshooting commands consist of the SuperUser command with WAN Extender parameters, and the SHow command with the normal WE Service parameters. This section describes the WAN Extender Service parameters that are used with the SuperUser command and shows the displays that they generate.

For a description of the normal -WE Service parameters, see the WE Service Parameters chapter in *Reference for Enterprise OS Software*; to see display samples, see “Sample Configuration Verification Displays” earlier in this chapter.

NETBuilder II troubleshooting commands allow qualified 3Com technicians to monitor, diagnose, or troubleshoot the WAN Extender and NETBuilder II bridge/router operations.



CAUTION: *The NETBuilder II troubleshooting commands for WAN Extender perform actions that may seriously impact the WAN Extender ability to accept and initiate calls. These commands should only be used by qualified 3Com support technicians or under their close supervision and only during periods of light or no-call traffic.*

The commands are grouped into the following categories:

- Tools for configuration verification

To verify that the downloaded configuration on the WAN Extender is correct, use:

```
SuperUser WanExtender !<WE-port> SystemInfo
SuperUser WanExtender !<WE-port> GlobalSystemParms
SuperUser WanExtender !<WE-port> NetPortParms
SHow !<WE-port> -WE ProFiles
SHow !<WE-port> -WE CONFIguration
```

- Tools for verification of correct operation

To verify that the NETBuilder II system in conjunction with the WAN Extender is operating as configured, use:

```
SuperUser WanExtender DisplayActiveConnections
SHow !<WE-port> -WE DevSTATistics
```

- Tools for problem diagnosis

To analyze and diagnose a problem when the NETBuilder II system or the WAN Extender are not operating as they were configured, enter:

```
SuperUser WanExtender TraceLevel
SuperUser WanExtender DlcITrace
```

- Tools for detailed debugging

The detailed debugging commands are reserved for use by engineers to debug problems that are difficult to analyze and diagnose with the other tools, and are not described in this chapter.

WAN Extender Service Parameters

This section provides a description, the syntax, and a display sample for the WAN Extender Service parameters, which are used with the SuperUser command.

SystemInfo This parameter retrieves and displays system type, memory configuration, and software version data from the attached WAN Extender using:

```
SuperUser WanExtender !<WE-port> SystemInfo
```

For example, enter:

```
SuperUser WanExtender !4 SystemInfo
```

A display similar to the following appears:

```
Global System Info parameters from WAN Extender out path 4:
isdn_version:
PRIS48M Rev5.20g 4/16/96 5.2.g
we_version:
WanExtender Rell.15E4 4/96
pcmcia_mem 524288 bytes
mem_size ..2097152 bytes
type .....WAN Extender 2T
```

GlobalSystemParms This parameter retrieves and displays the system-level parameters from the attached WAN Extender using:

```
SuperUser WanExtender !<WE-port> GlobalSystemParms
```

For example, enter:

```
SuperUser WanExtender !4 GlobalSystemParms
```

A display similar to the following appears:

```
Global System parameters from WAN Extender out path 4:
version .....0xA005
name .....NB2_4
clock source .....from Net Port 1
configured WE type ..WAN Extender 2T
baud .....4096 Kbps (local link)
max data pkt size ....1518 bytes
Console trace level ..NONE
lapb .....DISABLED
```

NetPortParms This parameter retrieves and displays the parameters configured for each network port of the attached WAN Extender using:

```
SuperUser WanExtender !<WE-port> NetPortParms
```

For example, enter:

```
SuperUser WanExtender !4 NetPortParms
```

A display similar to the following appears:

```
System parameters for Network Port 1 from WAN Extender out path 4:
call control .....ISDN
hunting .....ASCENDING
framing .....Extended Superframe
line code .....B8ZS
equalization .....0-133 ft from CSU
port digits .....986404
port digits number type ...Subscriber
port digits number plan ...ISDN
switch type .....AT&T 5ESS
service variant .....AT&T Custom
enabled bchannels .....0x007FFFFF
inverted HDLC .....Disabled
```

```
rate adapted .....Disabled
link termination type .....User Side
call types allowed .....56KB 64KB_Clear
call filtering .....Disabled
ISDN Low Level Parameters:
T200 .....DEFAULT USED
T203 .....DEFAULT USED
N200 .....DEFAULT USED
transmit_window ...DEFAULT USED
```

Network Port 2 on WAN Extender out path 4 is configured as UNUSED.

DisplayActiveConnections This parameter displays summary information related to currently active connections that have been established by the NETBuilder II system through the attached WAN Extender. Activate this parameter using:

```
SuperUser WanExtender DisplayActiveConnections
```

For example, enter:

```
SuperUser WanExtender DisplayActiveConnections
```

A display similar to the following appears (the first column values are virtual paths):

| | | | | | | |
|-----|--------|--------|---------|---------|---------|--------------------|
| V11 | path:4 | pid:4 | dlci:1 | kbps:64 | Nport:2 | channels:x00000001 |
| V56 | path:7 | pid:56 | dlci:32 | kbps:64 | NPort:2 | channels:x01000000 |
| V57 | path:7 | pid:57 | dlci:33 | kbps:64 | Nport:2 | channels:x02000000 |
| V58 | path:7 | pid:58 | dlci:34 | kbps:64 | Nport:2 | channels:x04000000 |
| V59 | path:7 | pid:59 | dlci:35 | kbps:64 | Nport:2 | channels:x08000000 |
| V60 | path:7 | pid:60 | dlci:36 | kbps:64 | Nport:2 | channels:x10000000 |
| v61 | path:7 | pid:61 | dlci:37 | kbps:64 | Nport:2 | channels:x20000000 |
| V62 | path:7 | pid:62 | dlci:38 | kbps:64 | Nport:2 | channels:x40000000 |

TraceLevel This parameter is used to turn on and off NETBuilder II bridge/router tracing on WAN Extender channels. Activate this parameter using (the <hex-mask> value is a bit mask that is used to indicate the types of tracing):

```
SuperUser WanExtender TraceLevel <hex-mask>
```

The following lists shows the hex values and the type of tracing each represents:

- 0x0000 disables tracing.
- 0x0001 traces unexpected execution paths.
- 0x0002 traces Simple Message Interface (SMI) messages exchanged between the NETBuilder II system and the WAN Extender. SMI messages are part of the WNI protocol.
- 0x0004 traces, with detailed information, SMI messages exchanged between the NETBuilder II system and the WAN Extender.
- 0x0008 traces the flow of messages through the system.
- 0x0010 provides a raw (hexadecimal) dump of all SMI messages received and sent by the NETBuilder II system.

To enable more than one type of tracing, the hex values corresponding to the types should be added and the resulting value specified for <hex-mask>.

For example, if you want to enable the following tracing types:

- Provide a raw (hexadecimal) dump of all SMI messages received and sent by the NETBuilder II system. (0x0010)
- Trace SMI messages exchanged between the NETBuilder II bridge/router and WAN Extender, (0x0004)
- Trace unexpected execution paths, (0x0001)

Enter:

```
SuperUser WanExtender TraceLevel 15
```

DlciTrace This parameter turns tracing on and off for all data and WNI frames on a single data link connection Identifier (DLCI) or all DLCIs currently established between the NETBuilder II system and the WAN Extender. DLCI traffic for multiple interfaces is displayed without differentiating the traffic of one interface from the other if more than one WAN Extender is connected to a NETBuilder II system.

Use this parameter only with one WAN Extender port enabled at a time or if DLCIs are not used by more than one WAN Extender interface. See “DisplayActiveConnections” earlier in this chapter to determine which DLCIs are in use.

To activate this parameter, use:

```
SuperUser WanExtender DlciTrace OFF | <dlci> | ALL
```

How the WAN Extender Works

A WAN Extender provides virtual paths to be used for interconnecting remote NETBuilder devices to a central site NETBuilder II bridge/router running PPP. The interconnection is established using channelized T1 and E1 leased-circuit services, and switched 56 and ISDN PRI switched-circuit services.

WAN Extender Models

There are two WAN Extender models:

- WAN Extender 2T

The WAN Extender 2T is intended for WAN networks that support the T1 interface. It provides two network interfaces, each of which can be independently connected to channelized T1, switched 56, or ISDN PRI services. The WAN Extender 2T supports 24 channels on each network interface for channelized T1 and switched 56 environments. It supports 23 B channels and one D channel on each network interface for ISDN PRI environments.

- WAN Extender 2E

The WAN Extender 2E is intended for WAN networks that support the E1 interface. It provides two network interfaces, each of which can be independently connected to channelized E1 or ISDN PRI services. The WAN Extender 2E supports 31 channels on each network interface for channelized E1 environments. It supports 30 B channels and one D channel on each network interface for ISDN PRI environments.

How Virtual Paths are Created

The WAN Extender provides an RS-530 connector (called the local port), which connects to a NETBuilder II bridge/router high-speed serial (HSS) RS-449 module, to provide a synchronous link between the two devices. A 3Com proprietary

interface protocol, called the WAN Extender/NETBuilder II Interface (WNI) Protocol, runs on this link.

The WAN Extender virtual paths are created automatically by the NETBuilder II bridge/router after it synchronizes with the WAN Extender over this link. Each virtual path can initiate a call to the WAN Extender and accept a call from the WAN Extender. There are three types of virtual paths: leased, DS0 dial, and H0 dial virtual paths.

Leased Virtual Paths

During the synchronization between the NETBuilder II bridge/router and the WAN Extender, the NETBuilder II bridge/router reads the profiles residing in the WAN Extender and sets aside a virtual path for each channelized T1 or E1 leased-line profile configured.

Although each virtual path is allotted one channel with a baud rate of 64 kbps when created, the channel expands to the size of the sum of all the channels specified in the profile when a connection is established.

The leased virtual paths occupy the bottom of the virtual path ID range.

DS0 Dial Virtual Paths

The virtual paths that are not used for leased lines are automatically available as dynamic paths in a dial-up path pool for interconnecting remote devices over switched ISDN or switch 56 lines.

The number of DS0 Dial virtual paths that are actually created is determined by the DialPathLimit setting, which considers the following information:

- The NETBuilder II bridge/router supports a maximum of 75 virtual paths.
- The number of virtual paths configured to be used for channelized leased lines.
- The number of virtual paths already configured for dial-up.
- The maximum number of channels that can be supported per port of the WAN Extender model being used (T1 supports 23 and E1 supports 30).

If the DialPathLimit setting is greater than the number of virtual paths that can be supported by the WAN Extender port, the number of virtual paths created will be the number of virtual paths supported, which is the smaller amount. For details on setting the DialPathLimit for DS0 virtual paths, see the WE Service Parameters chapter in *Reference for Enterprise OS Software*.

The DS0 Dial virtual paths occupy the top of the virtual path ID range.

H0 Virtual Paths

The number of H0 Dial virtual paths created is determined entirely by the value set for the H0 path count with the -WE DialPathLimit parameter. The range of H0 virtual paths is 0 to 3. Each H0 virtual path is 384 kbps, or equal to six DS0 dial virtual paths (6 x 64 kbps = 384 kbps). H0 and DS0 virtual paths can run on the same port at the same time.

For details on setting the DialPathLimit for H0, see the WE Service Parameters chapter in *Reference for Enterprise OS Software*.

For a complete description on ports and paths including how to number them, and for a description on how to set up ports and paths for a bridge/router using wide area interfaces, see the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

For information on setting up physical and virtual paths in dial-up pools for ISDN and switch 56 lines, see the Configuring Port Bandwidth Management chapter.

How the WAN Extender Operates

The WAN Extender is managed by an external software application called the WAN Extender Manager, which runs under Microsoft Windows 3.1 or later on a PC. The PC connects to the console port on the WAN Extender.

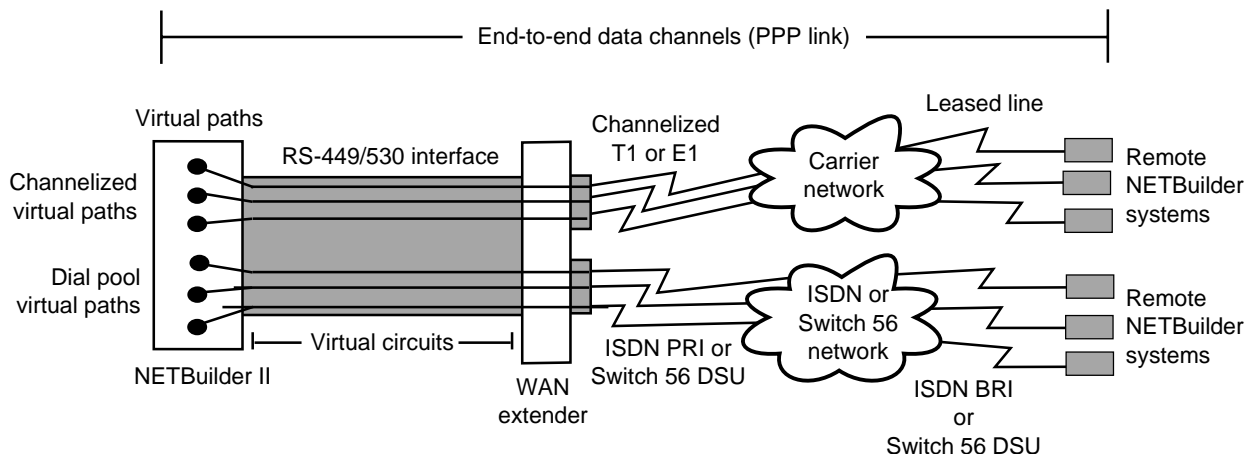
The WAN Extender maps each WAN connection to a data channel and makes that data channel available to the NETBuilder II bridge/router through a virtual path. The NETBuilder II bridge/router operates as follows:

- Views the data channel (a virtual path) as the underlying link for a virtual port
- Uses the data channel as if it were a clear channel
- Transparently establishes the end-to-end data link through the WAN Extender

When a connection to a remote site is first made using the WAN Extender, the end-to-end data link is established using PPP. After the end-to-end link is established, various higher-layer PPP NCP negotiations occur, depending upon your configuration at either end of the link, and then network layer protocol connection is established. Figure 331 shows the WAN Extender connection.

NETBuilder II bridge/router PPP-based virtual ports can be used to establish bridging, routing, and Boundary Routing connectivity. The Enterprise OS software operates the same way for WAN Extender virtual path-based ports as for any other point-to-point virtual port or port running PPP.

Figure 331 WAN Extender Connections to Remote NETBuilder Systems



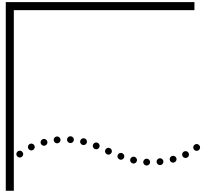
When a frame is received by the WAN Extender from the NETBuilder II bridge/router, frame forwarding proceeds as follows:

- 1 The WNI protocol header is stripped.
- 2 The data channel identifier is extracted from the WNI protocol header.
- 3 The data channel identifier is mapped to the corresponding network channel.
- 4 The frame is transmitted on the network channel.

When a frame is received by the WAN Extender on a network channel, frame forwarding proceeds as follows:

- 1** The network channel is mapped to the corresponding data channel.
- 2** The frame is prepended with a WNI protocol header containing a data channel identifier.
- 3** The frame is transmitted on the WAN Extender-to-NETBuilder II bridge/router connection.

When either the NETBuilder II bridge/router or the WAN Extender are initialized, or any time the link between the two systems is connected or enabled, the two systems engage in a local link synchronization process. On the NETBuilder II bridge/router side, the path associated with the WAN Extender connection goes to an UP state. After synchronization, the data channels may also get established, and the virtual port associated with each data channel may also go to an UP state. The data channel that gets established, and its associated virtual port that goes to an UP state, depends on the type of network the WAN Extender is connected to and the NETBuilder II bridge/router ports configurations.



CONFIGURING PORT BANDWIDTH MANAGEMENT

This chapter describes how to configure communication resources (telephone lines and digital circuits such as ISDN, T1/E1, and T3/E3 lines) for use with your dial-up wide area network (WAN) lines. In a 3Com WAN, you use *port bandwidth management* to control your communication resources. The concepts and configuration examples provided in this chapter will help you to decide how to use port bandwidth management to make effective and efficient use of your WAN communication resources.



Port bandwidth management is applicable to PPP WAN paths only.

Communication Resources Supported

Bandwidth management supports a broad range of communication resources, from public telephone lines using inexpensive analog modems to digital circuits providing throughput at rates up to 45 Mbps.

The virtual pipe can consist of any of the communication resources listed in Table 82.

Table 82 Communication Resources Supported by Bandwidth Management

| Resource | Throughput |
|--|--|
| Telephone line;
analog line using the public telephone system | Up to 56 kbps |
| Leased line;
dedicated higher quality analog line | Up to 56 or 64 kbps |
| ISDN BRI;
dedicated narrowband digital service | 56/64 or 112/128 kbps |
| ISDN PRI;
dedicated narrowband digital service | 23 or 30 channels at 64 kbps/56kbps |
| Switched-56;
nondedicated digital service | Up to 56 kbps |
| Fractional T1;
dedicated digital circuit | 64 kbps to 1.544 mbps in increments of 64 kbps |
| T1 or E1 line;
dedicated digital circuit | 24 or 30 channels at 64 kbps each |
| T1 or E1 channel;
dedicated digital circuit | 1.544 Mbps or 2048 kbps |
| T3 or E3 line;
dedicated digital circuit | Up to 45 Mbps |

“Configuring WAN Resources” later in this chapter provides instructions to configure these resources.

Sync Dial Lines When using sync dial lines (serial Data Terminal Equipment (DTE) dial-up lines), the bridge/router software supports the V.25 bis standard, which allows a DTE device to communicate with a Data Communication Equipment (DCE) modem or terminal adapter (TA). With V.25 bis, you can configure and store phone numbers in software for the modem or TA on the bridge/router. The phone number is sent to the bridge/router modem or TA when dialing occurs.

The software can also activate modems or TAs that store a phone number in the firmware of a data terminal ready (DTR) dialed modem or TA. DTR modems can only be used in static configurations; dynamic paths selected from a resource pool rely upon telephone numbers stored in the bridge/router software.

Integrated and Internal ISDN Lines The Integrated Services Digital Network (ISDN) interface is supported on model 42x, 43x, and 52x SuperStack II NETBuilder bridge/routers. These bridge/routers offer one basic rate interface (BRI) with two B channels (2B+D). ISDN is also supported on model 14x U and 14x S/T OfficeConnect NETBuilder bridge/routers.

WAN Extender Virtual Paths A WAN Extender virtual path is either ISDN 64 kbps/56 kbps, ISDN H0, switched-56, or channelized leased lines. Bandwidth management considers a WAN Extender virtual path as a generic path that can be allocated as a resource for the virtual pipe. See the Configuring the NETBuilder II to use a WAN Extender chapter for more information.

Associating Paths to Ports

3Com software uses the concepts of *ports* and *paths* to address interface connections. The basic interface connection is a port. A port is a logical interface that represents a connection to a network. The next logical connection is the path, which is the physical interface that connects the bridge/router to a physical medium such as an Ethernet local-area network (LAN), a token ring, or a serial line. In an ISDN environment, a path additionally represents the channel over which data is transmitted. This section describes the paths that can be configured on ports under bandwidth management.

Static versus Dynamic Paths You can unbind static paths from their ports and save them in a dial pool to be shared by more than one port. The paths in the dial pool are called *dynamic paths*. A path in the dial pool can be *dynamically bound* to a port or PPP virtual port when the path is needed to transfer data on a dial-up line. After the path is bound to the port, port-based dialing occurs. When the traffic is no longer present and the line is idle for a specified period, the path is unbound from the port and returned to the dial pool.

A port can bind to multiple dial paths. For each port, bandwidth can be dynamically allocated by bundling the multiple dial paths into the virtual pipe. Bandwidth can be allocated when and where it is most needed.

WAN Extender virtual paths are considered by bandwidth management to be generic paths that can be allocated to a logical or virtual port. Virtual paths are available and not tied to a specific physical resource until they are bound to a port. For virtual paths assigned to dial pools, the binding of the virtual path to a port occurs when an incoming call is received or when an outgoing call is started.

Dynamic Dial Path Pooling

You can use a dial pool to increase the reliability of your network configuration, achieve multideestination dialing by using dial phone number lists and modem pooling, and provide dynamic backup for leased or dial-up lines.

With multideestination dialing, you can allocate a small number of paths that are unbound from their ports to wait in the dial pool for an incoming call. You can create a PPP virtual port on the central router for each remote site and have all the virtual ports use the dial pool for path resources.

When the system receives an incoming call from a remote site, the dynamic path that answers is bound to a virtual port, which is standing by with the appropriate configuration information for the calling network. For the binding to occur, the identification of the remote site is transmitted to the central router.

For binding of dynamic paths to ports for all different types WAN services, the system caller ID (SCID) is used to identify the remote site NETBuilder bridge/router to a central site NETBuilder bridge/router.

The binding of dynamic paths for ports using SCID, which is configured with the `-SYS SysCallerID` parameter, can be used to identify and connect only 3Com NETBuilder bridge/routers at the remote site to the central site. Using SCID, the acceptance of an incoming call from an other-vendor bridge/router can only occur with a static path and port configuration.

For binding of dynamic paths or configuring static paths to ports for use with ISDN switched-circuit services, the remote site caller ID can also be specified with the Calling Line Identification Presentation (CLIP) dial string. The dial string is the ISDN phone number of the calling device.

Standard bundling on PPP links uses both the text endpoint discriminator (ED), and authentication. NETBuilder bridge/routers can provide a class 3, MAC address value for ED but they also accept classes 1 through 5 for ED from other systems. Authentication is achieved through PAP or CHAP. The bundle ID used by the NETBuilder bridge/router to identify links belonging to the same bundle is defined as a combination of ED and authentication. The ED is configured with the `-PPP TxEndpointDisc` parameter.

The binding of dynamic paths to ports to establish a connection between a remote site and a central site using CLIP, enables you to identify and connect 3Com NETBuilder bridge/routers and other-vendor bridge/routers at the remote site to the central site.

You can configure a port to identify remote site devices with only SCID or with only CLIP. If you have configured a port for CLIP and SCID, CLIP is used.

When you configure CLIP for a port, you add the remote site ISDN telephone number to the CLIP database using the `-PORT CLIPList` parameter and set the `-PORT DialRcvrState` parameter to `AnswerCLI`. See the `-PORT DialRcvrState` and the `-PORT CLIPList` parameters in the PORT Service Parameters chapter of *Reference for Enterprise OS Software* for details.

Because not all sites using a dial pool will be calling the central site at the same time, it is possible to share a small group of paths with a larger group of sites. Each site that can potentially call into the dial pool has its own virtual port predefined,

so there can be more virtual ports configured to use the dial pool than there are dynamic paths assigned to the dial pool. However, if all the remote sites dial the central router at the same time and only a small number of paths exist in the dial pool, some of the call attempts may fail due to a lack of path resources. These calls can be redialed at a later time.

For a summary of the terms used in this section, see “Bandwidth Management Terms” later in this chapter. For more information about the dial pool, see “Resource Aggregation” later in this chapter.

Valid Port and Path Configurations

Table 83 lists the valid combinations of port and path binding configurations available.

Table 83 Valid Port and Path Configurations

| Type of Path | Default Ports | Virtual Ports |
|-----------------------------------|---------------|---------------|
| Static leased line | 1 or more | Not supported |
| Static dial path | 1 or more | Not supported |
| Dynamic dial physical path | 1 or more | 1 or more |
| Dynamic WAN Extender virtual path | 1 or more | 1 or more |

System Bandwidth Management

Bandwidth management provides two operating modes: system bandwidth management and manual bandwidth management. These modes are enabled with the `-PORT DialInitState` parameter; the `DialOnDemand` option enables system bandwidth management mode, and the `ManualDial` option enables manual bandwidth management mode. This section describes system bandwidth management. See “Manual Bandwidth Management” later in this chapter for information about the manual bandwidth management mode.

System bandwidth management provides you with automated bandwidth management features. You provide bandwidth allocation guidelines and the system automatically manages the virtual pipe for you by monitoring traffic rates on the line. When traffic increases, bandwidth management may automatically allocate additional bandwidth; when traffic decreases, it may automatically decrease the bandwidth.

Changes to bandwidth and line characteristics take effect immediately. The system also uses a phone number list that you define to automatically obtain additional bandwidth. If bandwidth requirements consume more than one path, the system picks additional paths to form the virtual pipe and meet the requirements.

Dial-on-Demand

Dial-on-demand (DOD) is an economical way to use phone lines when communicating between bridge/routers. It is supported only under system bandwidth management mode and is enabled with the `DialOnDemand` option of the `-PORT DialInitState` parameter. DOD is triggered on when there is traffic on a port, and triggered off when the port is idle or experiences a decrease in traffic congestion. The careful monitoring of traffic provided by system bandwidth management allows more cost effective-use of DOD lines.

A connection is established when the system automatically dials a phone number specified with the `-PORT DialNoList` parameter, or the phone number configured in the DTR modem. The line stays up as long as transmit traffic is present. When

there is no more data, the call is terminated. It is automatically reestablished without any intervention when there is data to be sent across the line. Connections that are no longer in use are temporarily terminated until new demand occurs. The `-PORT DialIdleTime` parameter determines how long a connection must be idle before the call is terminated.

In general, DOD will limit background traffic to routed packets (DECnet, IP, and IPX-routed packets) and other network protocol packets (IPX RIPs and SAPs) that are absolutely necessary to maintain the functionality and integrity of the overall network. The software feature set provides you with the necessary parameters for controlling the traffic over that DOD link. For phone lines and the connections associated with those lines to operate properly in the DOD state, the network layer protocols running over those connections must use statically defined routes or Open Shortest Path First (OSPF) demand circuits (per RFC 1793). Currently IP, IPX, and DECnet are the only network layer protocols supported with DOD. For procedures and configuration examples of IP and IPX routing over a DOD link, see “Routing Configurations over DOD Links” later in this chapter. For information about routing DECnet over a DOD link, see the Configuring DECnet Routing chapter.

Bandwidth-on-Demand

Bandwidth-on-demand (BOD) is triggered on when the system detects traffic congestion on a port configured for system management mode. You specify the bandwidth that a port should operate at normally, then define the maximum amount of bandwidth above this setting that the port can have. Together these settings define the maximum width of the virtual pipe.

The BOD allocation strategy provides a flexible approach for configuring WAN dial-up lines. For example, the normal operating bandwidth of a WAN with two 64 kbps ISDN lines could be configured together for a total bandwidth of 128 kbps, or be configured at 64 kbps bandwidth with incremental increases up to 128 kbps, as traffic needs required. Depending upon traffic across your network, you can choose to configure one wide virtual pipe to handle the traffic, or configure a narrower virtual pipe that expands and contracts as traffic increases or decreases.

Bandwidth allocation is defined using the `-PORT NORMALBandwidth`, `BODThreshold`, `BODIncrLimit`, and `DialSamplPeriod` parameters, which specify bandwidth settings and the conditions that trigger BOD. Bandwidth management monitors the incoming and outgoing rate of traffic and uses the settings to prevent dropped packets by changing the size of the virtual pipe (allocating or removing lines and bandwidth) as required by traffic demands.

You can also configure a line as a general purpose line that can be allocated for any purpose, including disaster recovery, using the `UnRestricted` option of the `-PATH DialCONTROL` parameter. When a line failure causes the port bandwidth to drop below the level specified with the `-PORT NORMALBandwidth` parameter, the DOD strategy and bandwidth management work to restore the specified bandwidth. If traffic conditions warrant additional bandwidth, then BOD increases the bandwidth accordingly.

Disaster Recovery

Disaster recovery is the disaster recovery threshold, which is defined as the minimum of the normal bandwidth threshold, as defined by the `-PORT NORMALBandwidth` parameter, and the total amount of configured leased line bandwidth that is assigned to the port, excluding disabled paths.

When the total active bandwidth from the leased line paths falls below the disaster recovery threshold, bandwidth management tries to recover the port bandwidth to the target set with the `-PORT NORMAlBandwidth` parameter using dial paths. In this event, a path configured for disaster recovery is given preference. You configure a line specifically for disaster recovery using the `DisasterRcvry` option of the `- PATH DialCONTRol` parameter.

Path Configuration Summary

Table 84 summarizes the path configurations available with system bandwidth management.

Table 84 Path Configurations Available with System Bandwidth Management

| Strategy | Path Configurations Available |
|---------------------|---|
| Dial-on-demand | Single or multiple dial paths acting as a bundle |
| Bandwidth-on-demand | Single or multiple dial paths for bandwidth-on-demand bandwidth aggregation |
| Disaster recovery | Single or multiple dial paths for disaster recovery on all dial PPP ports |

Resource Aggregation

This section describes how bandwidth management finds additional WAN resources to aggregate for the virtual pipe.

Dial Number List

The software can select a phone number from a list of destination phone numbers associated with a port. The bridge/router can automatically select alternate phone numbers as backup if a previously dialed phone number does not result in a connection. The software knows whether the dial-up line is a static or dynamic path. If it is a static path, the software obtains a phone number from the dial number list specified with the `-PORT DialNoList` parameter and brings up the path. If DTR dialing is being used, the software uses the phone number stored in the modem.

If the dial-up line is a dynamic path, the software searches a dial pool to obtain an additional path. The software obtains a path from the dial pool, and binds it to the port. The software obtains a phone number for the dial number list specified with the `-PORT DialNoList` parameter. By using the dial number list, the software can try other phone numbers if the first number is unavailable. When searching the dial number list, the software seeks a path that matches the baud rate set for the path and will choose the path that matches or most closely matches that rate. By using the dial pool, the software can select another dynamic path if the first path is not working. When traffic conditions return to normal, the dial-up path is unbound from the port and returned to the dial pool.

Prioritized Path Preferences

You can prioritize dynamic and static dial paths for use by specific ports using the `-PORT PathPreference` parameter. No other port can use these reserved paths, unless these paths are reserved by more than one port. The software tries the preferred list of paths first before using path resources in the rest of the dial pool, and will seek static paths first. By specifying your path preference, you can reserve path resources for your dial-up lines, and ensure that a path is always available.

When you specify the selection sequence for a static path, you actually specify the priority sequence of selection for use by a port. A static path is inserted at the end of the path preference list by default. A static dial path with either the V.25 bis or DTR dial mode can be inserted in the list, but DTR does not use the dial number in

the DialNoList parameter to dial out; instead, it uses the dial number stored in the modem.

Leased line paths cannot be included in the path preference list because bandwidth management cannot bring a leased line up or down; leased line paths are brought up when the port is enabled.

Manual Bandwidth Management

Manual bandwidth management is enabled with the -PORT DialInitState parameter; the ManualDial option enables manual bandwidth management mode. Manual bandwidth management mode requires user intervention to control bandwidth on the line. You set a fixed amount of bandwidth using the -PORT NORMAlBandwidth parameter, and then issue the Dial command to bring up the line. Bandwidth management tries to meet the bandwidth specification, but does not monitor traffic or make any dynamic changes based on traffic rates. Bandwidth changes are only made according to what you specify with the -PORT NORMAlBandwidth parameter.

Manual Dial

Under manual bandwidth management, a line is brought up manually using the Dial command. The call remains connected until a timer expires or until you end the connection using the HangUp command. After the call has been disconnected, it can be reestablished only by issuing another Dial command.

The Dial command has a path mode and a port mode, and operation of this command differs depending upon the mode specified. The path-based Dial command is used mainly for testing and in event-based macros that automate line backup processes. See the description of this command in the Commands chapter in *Reference for Enterprise OS Software*.

The port-based Dial command manually dials on the specified port. The command accepts static or dynamic port numbers and an optional dial string. If a dial string is entered, the number must be listed in the -PORT DialNoList parameter. The call is placed on the available highest-priority phone number specified for the port using the -PORT DialNoList parameter. If the highest prioritized phone number is not available, the software tries to use the next phone number specified for the port, if more than one phone number is configured.

A telephone number must be inserted in the -PORT DialNoList parameter for dialing to occur, (although you can temporarily override the phone numbers in the DialNoList parameter by specifying a port and dial string with the port-based Dial command). To complete the call, the software automatically finds a path by first checking if a path is available in the path preference list. If one is, that path is used. If no path is available, the software determines whether the port can use the dial pool. If the port can use the dial pool, the software checks for an available path in the dial pool, binds it to the port, and makes the call.

Bandwidth management manages Dial command calls and makes bandwidth evaluations based on the -PORT NORMAlBandwidth setting. All ports configured for Dial must have a positive bandwidth setting. Bandwidth management aggregates bandwidth resources as needed to meet the NORMAlBandwidth setting, dialing for more resources if additional bandwidth is needed, or hanging up or substituting resources if less bandwidth is needed. You specify which phone number and path to select with the -PORT DialNoList and PathPreference parameters.

Manual Hangup The port-based HangUp command manually brings down all dial path resources and is the default. Under manual bandwidth management mode, the port is brought down when HangUp is issued unless there are leased lines active.

You can hang up calls on dynamic paths or static paths. If the path is dynamic and is currently bound to a port, the HangUp command disconnects the call, unbinds the path, and places the path back into the dial pool.

Manual Bandwidth Management Disaster Recovery Under manual bandwidth management, the software only maintains the normal operating bandwidth of a port; there is no monitoring and automatic allocation of additional resources if the line goes down. You can specifically configure disaster recovery for a leased line using the DisasterRcvry option of the -PORT DialCONTROL parameter. When disaster recovery is enabled, manual bandwidth management will restore the line bandwidth back to its original setting when a leased line fails and the port bandwidth falls below what has been specified for normal operation.

Bandwidth Management Status Displays You can monitor the dial path status, including the state of the ports under bandwidth management control, using the -PORT DialSTATUS parameter. The display from this parameter shows total port bandwidth, messages indicating congestion levels, and the intentions of the bandwidth manager for allocating additional resources.

Bandwidth Management Statistical Displays You can control costs by monitoring connection charges. A dial MIB to support this monitoring facility is provided. To obtain DOD statistics, see the Statistics Displays appendix.

Configuring Wide Area Networking Using Async PPP This section describes how to set up and configure wide area networking using the Async PPP.

Async PPP is used on paths whose LineType is Dial and whose DialMode is ATdial.

Configuration for Async PPP is mostly in the PATH service; the only PORT service configuration required for use of AsyncPPP is specification of PPP as PORT OWNER, and inclusion of Async types in the DialNoList.



LAPB will not operate over an AsyncPPP or AT dial path. As a result, history-based compression (COMPRESSType set to History on a port) cannot be used.

Configuring Async PPP and AT Dial This section describes how to configure Async PPP and AT Dial.

Prerequisites Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to Chapter 1 and Chapter 2.

Procedure To set up a path for Async PPP and AT dial, follow these steps:

- 1 Configure the path for dial with AT technology using:

```
SETD !<path> -PATH LineType = Dialup
SETD !<path> -PATH DialMode = ATdial
```

AT dial does not participate in auto startup; if LineType is left at AUto an async/AT modem is not detected.

- 2 Configure the external device type as Async so the port can distinguish it from non-async modem paths using:

```
SETD !<path> -PATH ExDevType = Async
```

- 3 Enable CTS flow-control so the modem can avoid buffer overflow by preventing the NETBuilder bridge/router from transmitting using:

```
SETD !<path> -PATH AsyFlowCtrl = CTS
```

- 4 Configure the baud rate for the path using:

```
SETD !<path> -PATH BAud = <baud rate>
```

Speeds supported for Async PPP paths are limited to the range of 1.2k bps through 38.4k bps; if a higher speed is configured, the system uses 38.4k, although no error message is displayed.

The baud rate specifies the speed of data between the modem and NETBuilder bridge/router, not the modem connect (carrier) speed. In general, this speed should be higher than the carrier speed (assuming the modem supports CTS flow-control); some modems enforce this.

The baud rate chosen must be supported by the modem and should probably be the lowest baud rate just above whatever the actual maximum modem (carrier) throughput is expected to be.

- 5 All of the preceding parameters go into effect when the path is (re)enabled using:

```
SETD !<path> -PATH CONTrol = Enable
```

When the path is configured for AT dial operation in this way, it attempts to initialize the modem whenever the path is enabled or the NETBuilder bridge/router notices that the modem has been (re)attached or power cycled. (This is determined by a drop then rise of the CTS input signal in command mode.)

Initializing the Modem

To initialize the modem, the NETBuilder bridge/router sends an initialization string, which is a configurable set of AT commands intended to configure the modem appropriately for use on the port. The modem initialization strings used are defined with the AsyMdmInitStr parameter, and assigned to a path with the AsyMdmName parameter.



The default initialization string is simply "AT" which presumes that the modem defaults (or is preconfigured) to the appropriate settings for operation with the NETBuilder bridge/router.

To define and assign a modem initialization string for a path, follow these steps:

- 1 Define an initialization string using:

```
ADD -PATH AsyMdmInitStr <name> "<string>"
```

where <name> is a token used to identify the string for use on specific paths, and <string> is the AT command string used to configure the modem. The <name> consists of sixteen (16) characters, which can be letters, digits, underscore (_),

hyphen (-), period (.), or asterisk (*); the first character must be a letter. The <string> must be enclosed in double quotes, and should begin with AT. For example, enter:

```
ADD -PATH AMIS courier "AT&F1 E0V0Q0X0&A0 &C1&D2&R1 S0=0 &C1&D2"
```



Although the parameter accepts up to 80 characters of printable ASCII as the string definition, NETBuilder software version 10.1 only allows a maximum of 50; if a larger initialization string is defined, it will not be used and the path will not come up. In addition, the size of the modem command buffer may further reduce the available length of the initialization string.

The initialization string should configure the modem as needed; configuration requirements for a modem to operate with the NETBuilder bridge/router include:

- No command echo
- Numeric result codes
- Minimal result codes set
- Auto-answer disabled
- Character format 8/N/1
- Software flow-control (XON/XOFF) disabled
- Hardware flow-control (CTS) enabled
- Fixed-speed DTE link (doesn't follow carrier)
- CD output signal asserted according to carrier
- DTR drop during connection causes hang-up
- CTS asserted when in command mode

See "Modem Initialization Strings" later in this chapter for more specifics on each item, and some information on determining an appropriate initialization string to meet NETBuilder bridge/router and modem compatibility requirements.

2 Specify which predefined initialization string to use on the path using:

```
SETD !<path> -PATH AsyMdmName = <name>
```

where <name> is one of the names in the AsyMdmInitStr table, defined in step 1 above. If <name> is not specified or if ? is used, the NETBuilder help displays the currently defined names; AsyMdmInitStr table can also be displayed with the SHow command. (The reserved name "none" removes any assignment, so the path will use the default initialization string.)

If an undefined <name> is assigned to the path, a warning message is issued but the assignment is allowed to take effect. The NETBuilder bridge/router uses the default initialization string on that path until a definition for <name> exists. Also, <name> is case-insensitive; for example, the following command refers to the name "courier" defined in step 1 above:

```
SETD !3 -PATH AMN = Courier
```

The path must be (re)enabled for the new assignment to take effect:

```
SETD !<path> -PATH CONTrol = Enable
```

Adding a Number to the Phone Number List

When the path has been configured for AT dial and Async PPP operation, it is available to originate or answer calls for a PPP port. For a PPP port to use the AT

dial path to make a call, it must have a phone number in its DialNoList with an assigned type of Async.

- 1 To include add an async number to the phone number list, use:

```
ADD !<port> -PATH DialNoList "<phone-no>" [Baud=<baud>] Type=Async [Pos=<1-xx>]
```

The default baud rate attribute for an Async path is 9.6.

The string specified as the phone number MUST NOT include the AT prefix or the D (dial) command; when attempting to dial, the NETBuilder bridge/router adds the prefix ATDT (or ATDP) to the specified string and sends it to the modem. The first character of the string can be used to specify tone (T) or pulse (P) dialing.

See the description of the DialNoList parameter in *Reference for Enterprise OS Software* for more information.

Other port and path dial-related parameters are configured in the same way as for synch paths; see *Reference for Enterprise OS Software* or *Using Enterprise OS Software* for more information.

Modem Initialization Strings

For the NETBuilder bridge/router to make use of an AT dial modem, a number of conditions are imposed on modem operation. The commands required for the modem to operate this way can be used as an "initialization string" (assigned using the PATH AsyMdmInitStr and AsyMdmName parameters) that the NETBuilder bridge/router sends to the modem whenever the path is enabled or the modem is (re)attached or power cycled.



Some modem initialization may be required offline to put the modem into a state where the NETBuilder bridge/router will send it commands and it will accept them; in particular, CTS must be active and CD inactive for the NETBuilder bridge/router to send the initialization string.

The modem must accept an incoming call and dial an outgoing call using the A and D commands, respectively. In addition, the modem must be configured as follows:

- No command echo
- Numeric result codes
- Minimal result codes set
- Auto-answer disabled
- Character format 8/N/1
- Software flow-control (XON/XOFF) disabled
- Hardware flow-control (CTS) enabled
- Fixed-speed DTE link (does not follow carrier)
- CD output signal asserted according to carrier
- DTR drop during connection causes hang-up
- CTS asserted when in command mode

These requirements are grouped roughly into three categories for more detailed discussion below. In some cases, specific modem commands may be referenced; you may need to consult your modem technical documentation to determine if a particular command is supported (and/or required) for your modem.

Recommended initialization strings for some modems can be found at the end of this section.

Command Interaction The NETBuilder bridge/router sends commands to the modem in order to configure it (using the initialization string), place a call, or accept a call.

Commands should not be echoed; the only command-mode output expected from the modem is a single-digit response indicating the result of an operation:

- 0 -- OK (command successful)
- 1 -- CONNECT (carrier established)
- 2 -- RING (incoming call indication)
- 3 -- NO CARRIER (exiting call failed)
- 4 -- ERROR (command failed)

The modem must be configured so that automatic answer mode is disabled. When an incoming call arrives, the modem should signal the NETBuilder bridge/router using a code 2 (RING); if the bridge/router chooses to answer the call, it will respond with an ATA (answer) command.

This configuration can be achieved using basic AT commands which are fairly standard across a number of modems, although defaults tend to vary:

- E0: Do not echo commands
- Q0: Display result codes
- V0: Use numeric result codes
- X0: Minimal result code subset
- S0=0: Disable auto-answer mode



Some modems may have additional result code commands that may need to be used to restrict the result code subset. You may want to use other command settings relating to modem command interaction, for example disabling local/remote escape sequences.

Signal Interaction When an AT dial path is enabled, the NETBuilder bridge/router asserts DTR and RTS. When connected, the bridge/router indicates a hang-up request by toggling the DTR signal. (The RTS signal is toggled only when switching between command mode and data mode, and can be ignored.)

The NETBuilder bridge/router uses CTS for modem presence in command mode. Commands are only sent if CTS is asserted; a CTS toggle restarts the sequence. In data transfer mode, however, CTS may be used for flow-control. The NETBuilder bridge/router expects CD (modem carrier) only in response to an answer or dial command; it should drop to indicate hang-up. The only restriction on DSR is that it not drop in command mode (but it may follow CD).

Signal control commands appear to be somewhat less consistent across different modems, but the following commands apply to several:

- &C1: CD according to carrier
- &D2: DTR drop initiates hang-up
- &S0: DSR remains asserted

These commands may vary slightly on a per-modem basis, and some modems may have several commands that interact to control a single signal. Flow-control options are even more varied, and are covered in "Online Operation" next.

Online Operation

Most online modem operation is independent of the NETBuilder bridge/router. Choices may depend on the quality of your telephone service and the type of modem(s) you will be communicating with. The following lists some of the constraints to keep in mind:

- DTE speed must not follow carrier speed
The NETBuilder path speed is determined by the BAud parameter. Some modems automatically change their DTE speed based on carrier speed, particularly when answering a call. If the modem and NETBuilder bridge/router use different speeds the connection fails; it is essential that the modem maintain a constant DTE speed -- for example, based on the last AT command.
- Character format of 8/N/1.
The NETBuilder bridge/router uses 8 databits and no parity on an AT dial path; if the modem does not automatically pass this transparently it must be configured to do so. (Most modems learn the character format from the AT characters themselves.)
- Software flow-control (XON/XOFF) should be disabled.
The NETBuilder bridge/router does not use XON/XOFF, and during LCP option negotiation it always proposes an ACCM of zeros. Thus any XON/XOFF characters in data must be able to traverse the modem link as is.
- Hardware flow-control (CTS) may be enabled.
If the AsyFlowCtrl parameter is set to CTS, the NETBuilder bridge/router will not transmit data when CTS is low; this may improve operation by preventing overrun of the modem buffer. Most modems allow some form of hardware flow-control (some with restrictions) to take advantage of this. (The modem need not honor RTS flow-control; the NETBuilder bridge/router does not use it, and simply keeps RTS asserted.)
- Hang-up delay disabled.
The NETBuilder bridge/router assumes the modem is available again after dropping DTR for a short time. With an error-control in use, some modems delay a hang-up initiated by DTR drop if there is buffered data to deliver. In the NETBuilder environment, much of the value of this extra delay is lost, and it should be minimized if possible. Many modems use S-Register 38 to configure this delay.
- Escape sequences disabled.
The NETBuilder bridge/router does not use escapes to enter modem command mode. Many modems require "guard time" in escape sequences; Async PPP data from the NETBuilder bridge/router is explicitly framed, making chances of an inadvertent escape negligible. But if a modem allows escape sequences without a guard time, this may be a concern.
- Lower bound on carrier speed.
Some modems offer the capability to negotiate carrier no lower than a specified "floor", or reject connection if signal quality is too low. This may be desirable to prevent excessively slow links when a connection is of unusually poor quality. (In fact, if a connection is too slow the PPP link may not come up at all.)

In general, the configuration commands for these options vary widely between different modem manufacturers and models.

Example Initialization Strings

Initialization strings for several modems successfully used with the NETBuilder bridge/router are listed in Table 85. These strings generally show minimum configuration required; additional configuration may be desirable (or even required) for a particular environment.

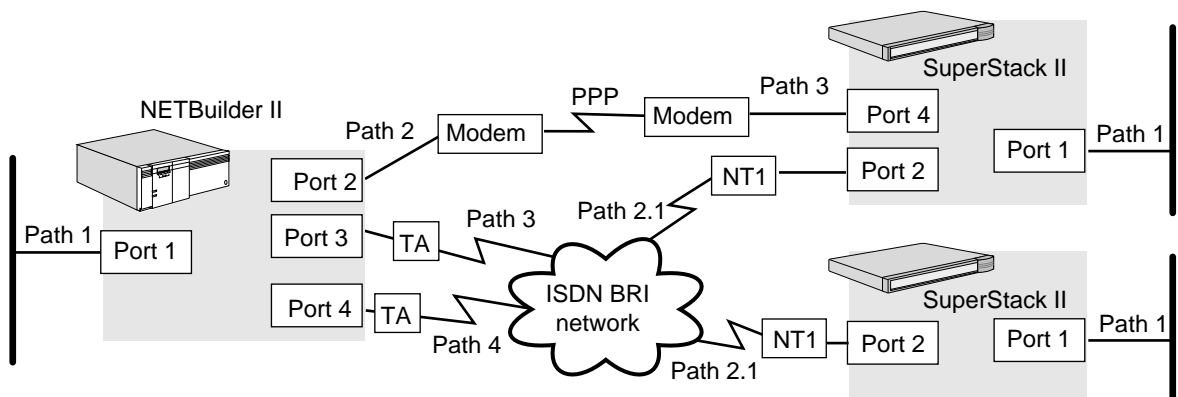
Table 85 Modem Initialization Strings

| String | Modems |
|---|-------------------------------|
| "AT&FQ0V0E0&C1&D2&S0X0S0=0" | Hayes (Accura 33.6) |
| "AT&FQ0V0E0&C1&D2&S0X0S0=0S2=128S38=0" | Motorola (ModemSURFR 33,600) |
| "AT&F1E0V0Q0X0&C1&D2S0=0&R1&A0" | 3Com/USR (Courier, Sportster) |
| "AT&F8&W0&FE0V0Q0X0&Q1&C1&D2&E4&E1&E12%E0S0=0S13=0" | Multitech (MT1932ZDX) |

Configuring WAN Resources

The procedures in this section prepare your WAN resources for use with bandwidth management. See Figure 332 for an illustration of the configuration examples in this section.

Figure 332 Basic Bandwidth Management Port and Path Configurations



Configuring Dial-Up Lines Using a Modem or TA

You can use digital ISDN and analog serial lines to establish connectivity with remote sites so that these sites can send updates to a central location. You can also configure serial lines when using Boundary Routing software.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Procedure

To configure a DTE serial line or ISDN line with a TA to use with bandwidth management, follow these steps:



You must complete this procedure at both ends of the link.

1 Set up the line type.

- a** By default, the dial path is set to Auto for the SuperStack II boundary router. You can check that the setting has not been changed using:

```
SHow !<path> -PATH LineType
```

- b** To change the value of the parameter to Dialup, use the following syntax:

```
SETDefault !<path> -PATH LineType = Dialup
```

Use this syntax to select Dialup as the line type for the NETBuilder II bridge/router.

2 Specify the baud rate for the device.

- a** Set the baud rate for a serial line using:

```
SETDefault !<path> -PATH Baud = <kbps> (1.2-52000)
```

The default baud rate for a serial line is 64 kbps. The auto startup feature automatically detects modem connections on the SuperStack II bridge/router; it does not sense the baud rate for ISDN paths with external TAs attached.

- b** Set the baud rate for an ISDN line connected with a TA using:

```
SETDefault !<connectorID> -PATH Baud = <kbps> (1.2-52000)
```

The auto startup feature automatically detects modem and TA connections on the SuperStack II bridge/router. See "Configuring ISDN Lines" next for more information about configuring ISDN lines without a TA.

3 Select the connector type using:

```
SETDefault !<path> -PATH CONNector = V35 | RS232 | RS449 | G703 | HSSI | X21
```

This step is not required when using FLEXWAN ports.

4 Set the transmit clock for the bridge/router using:

```
SETDefault !<path> -PATH CLock = TestMode | External | Internal
```

The Internal value applies to model 32x and 52x SuperStack II bridge/routers only. The External value allows the bridge/router to derive the transmit clock from either the send or receive timing clock supplied by the digital service unit/channel service unit (DSU/CSU) or by the attached modem.

This step is not required when using FLEXWAN ports.

5 Select either the V.25bis standard or DTR dialing mode using:

```
SETDefault !<path> -PATH DialMode = V25bis | DTRdial
```

Select V.25 bis to configure a DTE serial line using a V.25 bis-compatible modem. Select DTR dial to configure a line using a modem that uses the DTR signal to initiate a call.

If you are using the V.25 bis standard, specify the telephone number of the remote site being dialed using:

```
ADD !<port> -PORT DialNoList "<phone-no>"
```

For more information, see "Configuring the Dial List" later in this chapter.

6 Specify the external device type attached to the DTE path using:

```
SETDefault !<path> -PATH ExDevType = [Modem | Bri | Sw56]
```

Port-based dialing that uses phone numbers from the dial-number list always looks at the setting of the ExDevType parameter to select an appropriate path for the phone number and phone technology. The default setting of the ExDevType parameter is Modem; the default setting for the Type attribute of the DialNoList parameter is also Modem.

7 Set the path characteristics for the line using:

```
SETDefault !<path> -PATH DialCONTRol = ([DYNamic | STAtic], [DisasterRcvry
| NoDisasterRcvry | UnReSTRicted])
```

The -PATH DialCONTRol parameter provides several options for setting the line.

The STAtic value allows the selected path to be statically bound to its corresponding port and is the default. The DYNamic setting unbinds a path from its corresponding port and adds the path to the dial pool. A static path is not part of the dial pool. Placing a path in the dial pool allows the path to be used by any dial port. For a dial path to become a dynamic dial path, the -PATH LineType parameter must be set to Dialup.

You can also choose to set the line specifically for disaster recovery, or as unrestricted to allow it to be used for any purpose including disaster recovery. The NoDisasterRcvry option prevents the line from being used for disaster recovery and is usually assigned to the slowest or least reliable line on the network.

For example, to configure the analog serial line on path 3 for no disaster recovery, enter:

```
SETDefault !2 -PATH LineType = Dialup
SETDefault !2 -PATH Baud = 28.8
SETDefault !2 -PATH CONNector = RS232
SETDefault !2 -PATH CLock = External
SETDefault !2 -PATH DialMode = DTRdial
SETDefault !2 -PATH ExDevType = Modem
SETDefault !2 -PATH DialCONTRol = NoDisasterRcvry
```

8 Enable the line and make sure all settings on the path take effect using:

```
SETDefault !<path> -PATH CONTRol = Enabled
```

Configuring ISDN Lines

You can use ISDN lines to establish connectivity with remote sites so that these sites can send updates to a central location. See Figure 332 for an illustration of the configuration examples in this section.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Acquire services from a telecommunications carrier.

Procedure

To configure an ISDN line for use with bandwidth management, follow these steps:

- 1 To set the line type to Dialup for SuperStack II bridge/routers with an ISDN interface, enter:

```
SETDefault !2.1 -PATH LineType = Dialup
```

- 2 Set the switch type.

By default, the switch type is set to European Telecommunications Standards Institute (ETSI). If you need to change the switch type setting, use:

```
SETDefault !<connectorID> -PATH SwitchType = ETSI | NIT | KDD | NI1 |
ATT5ESS | DMS100 | VN3 | AUSTEL
```



ETSI is the default and is only for users in the United Kingdom and Germany. See the PATH Service Parameters chapter in Reference for Enterprise OS Software to determine which switch type settings are supported and how international users should configure this parameter.

- 3 Specify a local telephone number using:

```
SETDefault !<connectorID.channelID> -PATH LocalDialNo = "<string>"
```

- 4 If you are planning to use an additional channel as a backup line and your telecommunications carrier provided only one telephone number for all channels, specify a subaddress using:

```
SETDefault !<connectorID.channelID> -PATH LocalSubAddr = "<string>"
```

When specifying the subaddress, you can specify up to 20 ASCII characters. See the Configuring Wide Area Networking Using ISDN chapter for information on why you would set up a subaddress.

You can also specify the telephone number of the remote site being dialed by using the -PORT DialNoList parameter. The phone number usually includes the dial prefix, country code, area code, and possibly a subaddress assigned to your ISDN interface. If you specify a subaddress, you must separate the phone number from the subaddress with a semicolon (;). With ISDN phone numbers, you can use hyphens (-) to separate the prefix, country code, and area code. For more information, see "Configuring the Dial List" later in this chapter.

- 5 Set the path characteristics for the line using:

```
SETDefault !<path> -PATH DialCONTRol = ([DYNamic | STATic], [DisasterRcvry
| NoDisasterRcvry | UnReSTRicted])
```

The -PATH DialCONTRol parameter provides several options for setting the line.

The STATic value allows the selected path to be statically bound to its corresponding port and is the default. The DYNamic setting unbinds a path from its corresponding port and adds the path to the dial pool. A static path is not part of the dial pool. Placing a path in the dial pool allows the path to be used by any dial port. For a dial path to become a dynamic dial path, the -PATH LineType parameter must be set to Dialup.

You can also set the line specifically for disaster recovery, or set it as unrestricted to allow it to be used for any purpose including disaster recovery. The NoDisasterRcvry option prevents the line from being used for disaster recovery and is usually assigned to the slowest or least reliable line on the network.

For example, to configure ISDN on path 2.1 as an unrestricted, dynamic line, enter:

```
SETDefault !2.1 -PATH LineType = Dialup
SETDefault !2.1 -PATH SwitchType = ATT5ESS
SETDefault !2.1 -PATH LocalDialNo = "1-213-555-1212"
SETDefault !2.1 -PATH LocalSubAddr = "100"
SETDefault !2.1 -PATH DialCONTRol = (DYNamic, UnReSTRicted)
```

- 6 If you are configuring ISDN for North American BRI ISDN dial-up modes, specify the Service Profile Identifiers (SPIDs) using:

```
SETDefault !<connectorID> -PATH SPIDdn1 = "<string>"
```

Some North American ISDN switches require two SPIDs. In this case, you will need to add the SETDefault !<connectorID> -PATH SPIDdn2 = " <string>" parameter to your configuration. For DMS 100, the string must contain a Service Profile Identifier (SPID) and a directory number (DN) separated by a semicolon (;).

For example, to set the SPID to 4085551212 and the DN to 1234567, enter:

```
SETDefault !2.1 -PATH SPIDdn1 = "4085551212;1234567"
```

- 7 Enable the line and make sure all settings on the path take effect using:

```
SETDefault !<path> -PATH CONTRol = Enabled
```

- 8 If you changed the switch type or a SPID parameter, you must reboot the system for the changes take effect.

Configuring Leased Lines

You can use leased lines to establish connectivity with remote sites so that these sites can send updates to a central location.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Procedure

To configure a leased line to use with bandwidth management, follow these steps:



You must complete this procedure at both ends of the link.

- 1 Set the line type to Leased.
 - a By default, the dial path is set to Leased for all NETBuilder II systems. You can check that the setting has not been changed using:

```
SHow !<path> -PATH LineType
```

- b To change the value of the parameter, use:

```
SETDefault !<path> -PATH LineType = Leased
```

- 2 Specify the baud rate for the device using:


```
SETDefault !<path> -PATH Baud = <kbps> (1.2-52000)
```
- 3 Select the connector type using:


```
SETDefault !<path> -PATH CONNector = V35 | RS232 | RS449 | G703 | HSSI | X21
```

This step is not required when using FLEXWAN ports.

- 4 Set the transmit clock for the bridge/router using:

```
SETDefault !<path> -PATH CLock = TestMode | External | Internal
```

The Internal value applies to model 32x and 52x SuperStack II bridge/routers only. The External value allows the bridge/router to derive the transmit clock from either the send or receive timing clock supplied by the digital service unit/channel service unit (DSU/CSU) or by the attached modem.

This step is not required when using FLEXWAN ports.

- 5 Enable the line and make sure all settings on the path take effect using:

```
SETDefault !<path> -PATH CONTrol = Enabled
```

Configuring ASPC Leased Lines

You can configure an ISDN interface for the Austel Semi Permanent Circuit (ASPC) line type when this line type is supported by the ISDN switch in use. ASPC is a leased line type that is controlled like a dial line but operated as a leased line. Unlike other leased lines, a phone number is associated with the ASPC line.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

Procedure

To configure an ASPC leased line to use with bandwidth management, follow these steps:



You must complete this procedure at both ends of the link.

- 1 Set the line type to ASPC.
- a By default, the dial path is set to Leased for all NETBuilder II systems. You can check that the setting has not been changed using:

```
SHow !<path> -PATH LineType
```

- b To change the value of the parameter, use:

```
SETDefault !<path> -PATH LineType = ASPC
```

- 2 Set the ASPC number using:

```
SETDefault !<path> -PATH ASPCNumber "dial string"
```

The "dial string" may contain up to 52 alphanumeric characters, and it must be enclosed in double quotation marks. "Dial string" usually contains the ISDN phone number of the ASPC switch. If the dial string is empty (two double quotation marks only are present), the existing number is deleted. The maximum number of entries per path is one.

- 3 Enable the line and make sure all settings on the path take effect using:

```
SETDefault !<path> -PATH CONTrol = Enabled
```

Configuring System Bandwidth Management Mode (DOD)

This section describes the procedure to enable system bandwidth management mode.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Enable your WAN communication resources according to “Configuring WAN Resources” earlier in this chapter.

Procedure

To enable system bandwidth management mode, set the initiator state to enable system bandwidth management mode using:

```
SETDefault !<port> -PORT DialInitState = DialOnDemand
```

This command also enables dial-on-demand.

Configuring Bandwidth-on-Demand

This section describes how to enable bandwidth-on-demand for the system bandwidth management mode. See Figure 332 for an illustration of the configuration example in this section.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Enable your WAN communication resources according to “Configuring WAN Resources” earlier in this chapter.

Procedure

To enable bandwidth-on-demand (BOD), follow these steps:

- 1 Specify the amount of bandwidth the port should bring up when enabled using:

```
SETDefault !<port> -PORT NormalBandwidth = <kbps>
```

- 2 Enable BOD and specify the amount of additional bandwidth that bandwidth management can allocate for a port using:

```
SETDefault !<port> -PORT BODIncrLimit = <kbps>
```

This syntax specifies incremental bandwidth levels that can be allocated for the port.

- 3 Specify the conditions that trigger additional path resources for BOD using:

```
SETDefault !<port> -PORT BODThreshold = <%>(0-100)
```

- 4 Specify the amount of time bandwidth management should wait to take action to bring a port up or down using:


```
SETDefault !<port> -PORT DialSamplPeriod = <seconds>(0-300), (0-300)
```

For example, enter the following commands configure a DOD line with a normal port bandwidth of 64 kbps:

```
SETDefault !3 -PORT DialInitState = DialOnDemand
SETDefault !3 -PORT NormalBandwidth = 64
SETDefault !3 -PORT BODIncrLimit = 64
SETDefault !3 -PORT BODThreshold = 50
SETDefault !3 -PORT DialSamplPeriod = 30, 60
```

Traffic must exceed 32 kbps for 30 seconds before bandwidth management brings up additional bandwidth using BOD. The additional dial path is taken down when the rate of traffic is less than 32 kbps for longer than 60 seconds.

Configuring the Dial List

A dial list allows the software to select a phone number from a list of destination phone numbers associated with a port. Numbers in the dial list are selected sequentially. The software provides options for ordering the numbers in this list.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Enable your WAN communication resources according to “Configuring WAN Resources” earlier in this chapter.

Procedure

To configure a list of telephone numbers to dial for remote WAN connections, and as possible resources for additional bandwidth allocation, follow these steps:

- 1 To allow your bridge/router to dial out, configure the dial number list using:

```
ADD !<port> -PORT DialNoList "<phone no>" [Baud = <rate> (1.2-16000)]
  [Type = Modem | Bri | Sw56 | WE | WEH0] | [Pos = <number>]
```

You can enter this command more than once to append a phone number or profile to the dial list.

The string entered for the WAN Extender profile is case-sensitive, can contain alphanumeric characters, and can be no longer than 52 characters.

With ISDN phone numbers, you can use hyphens (-) to separate the prefix from the country code from the phone number. For ISDN, the phone number includes a dial prefix, country code, and area code and possibly a subaddress. If you specify a subaddress, you must separate the phone number from the subaddress with a semicolon (;). You can configure up to 16 phone numbers per port.

For V.25 bis dialing, the phone number can include a dial prefix, country code, and area code.



The software uses the Baud and Type keywords to make a path match. It is important to enter the same device type as you entered in the -PATH ExDevType parameter; software will make a best match for the baud rate. See “Configuring WAN Resources” earlier in this chapter for use of these commands.

- 2 You can also specify the number of time the software attempts to redial the remote system if the call attempt fails using:

```
SETDefault !<port> -PORT DialRetryCount = <number> (0-20)
```

If dialing is based on a static port and path binding, the software first tries to make the call. If the attempt fails to bring the path up, the software tries the call again using the same or different path. The call attempts continue until the dial retry count is reached.

You can also append phone numbers to the end of the list, insert a phone number into a specific position in the list, edit an existing phone number, or delete an existing phone number. To add a phone number into a specific position in the list, see "Adding a Phone Number" next.

Example 1 To enter a Los Angeles phone number for port 2 that consists of a long-distance dial prefix 1 (assume that the bridge/router being configured is located in Santa Clara), the phone number 213-456-7000, and the subaddress 101, enter:

```
ADD !2 -PORT DialNoList "1-213-456-7000;101" Baud = 56 Type = Bri
```

Example 2 The DialNoList parameter includes options for WAN Extender virtual paths. To add a dial number for virtual port V1 and instruct the system to go to port 4 with WAN Extender (WE) profile 5, enter:

```
ADD !V1 -PORT DialNoList "4 5" Type = WE
```

Example 3 To add a London phone number for port 2, at the end of a dial-up phone list, that consists of the international dialing code 011, the U.K. country code 44, and the phone number 213-456-7000, enter:

```
ADD !2 -PORT DialNoList "011 44 213 456 7000"
```

This entry ignores the Baud rate, Type, and Pos (position on the list).

For DTR dialing, the phone number is irrelevant, because the outgoing telephone number is stored in the modem.

Example 4 You can configure the dial number list to dial the same number repeatedly by adding multiple copies of the number. Prefix the phone number with a variable number of periods to distinguish the duplicate entries by entering:

```
ADD !V1 -PORT DialNoList "123 4567"
ADD !V1 -PORT DialNoList ".123 4567"
ADD !V1 -PORT DialNoList "..123 4567"
```

The bridge/router dials 123-4567 three times. This technique works for both ISDN and analog phone numbers.

Adding a Phone Number

To insert a phone number into a specific position in the dial number list, enter the Pos (Position) keyword with a non-zero number after the dial string.

For example, to insert a phone number for port 4 into position 2 of the dial number list that contains 10 phone numbers, enter:

```
ADD !4 -PORT DialNoList "510 555 7000" Pos = 2
```

The software inserts the new phone number into position 2. The phone number that was previously in position 2 is now in position 3. If the phone already exists in the dial number list, it will be moved to position 2. You can insert the same phone

number twice by using blanks or other non-dialing characters. You also can include the Baud and Type keywords in any order when inserting phone numbers into the dial number list.

Editing an Existing Phone Number

To edit an existing phone number in the dial number list, you can change the position in the list, change the baud rate, and change the device type.

For example, if port 3 has already been assigned 612-345-3989 in position 2 with a baud rate of 64 kbps, you can change the baud rate by entering:

```
ADD !3 -PORT DialNoList "612 345 3989" Pos = 2 Baud = 14.4
```

Because the dial string is case-sensitive, make sure to match it exactly to successfully edit an existing string when characters other than numbers are used.

Deleting a Phone Number

To remove a phone number or profile from the dial number list, use:

```
DElete !<port> -PORT DialNoList "<phone no>"
```

The profile name is case-sensitive and must be matched exactly to be deleted.

Binding Paths to Ports

3Com software uses the concept of port and path bindings to pair a logical interface (port) to a physical network resource (path) such as an ISDN line. A port can bind to multiple dial paths. For each port, the bandwidth can be dynamically allocated by bundling the multiple dial paths into the virtual pipe. This concept allows bandwidth to be allocated when and where it is most needed.

The software also allows you to create dynamic paths by unbinding static paths from their ports and saving them in a dial pool to be shared by more than one port. A path in the dial pool can be bound to a port when the path is needed for data transfer events associated with dial-up.

The procedures in this section illustrate how to create a dynamic path, how to convert the dynamic path back to a static path, and how to identify remote site to a central site using Calling Line Identification Presentation (CLIP) so that the paths can be bound to a port.

Converting a Static Path to a Dynamic Path

Paths (except WAN Extender paths) are static by default. To convert a static path to a dynamic path, follow these steps:

- 1 Unbind a path from its corresponding port and add the port to the dial pool by entering:

```
SETDefault !1 -PATH DialCONTRol = DYNameic
```

- 2 Enable the path by entering:

```
SETDefault !1 -PATH CONTRol = Enabled
```

Changing a Dynamic Path to a Static Path

To change a dynamic path to a static path and remove it from the dial pool, follow these steps:

- 1 Convert a dynamic path to static by entering:

```
SETDefault !1 -PATH DialCONTRol = STATic
```

- 2 Reassign a path or multiple paths to a port using:

```
ADD !<port> -PORT PATHs <path>
```

When a static dial path is added to a port, it is automatically inserted at the end of the path preference list; see the next section for further information about the path preference list.

- 3 Enable the path by entering:

```
SETDefault !1 -PATH CONTROL = Enabled
```

Identifying Remote Sites with SCID and CLIP

To bind a dynamic path to a port to connect a remote site bridge/router to a central site, the remote site bridge/router must be identified.

If you are connecting a remote site bridge/router to a central site bridge/router over ISDN or non-ISDN lines, you can use the system caller ID (SCID) to identify the remote site bridge/router to the central site bridge/router only if the remote site bridge/router is a 3Com NETBuilder bridge/router. The SCID for a remote site can be a telephone number or a text string.

If you are connecting a remote site bridge/router to a central site bridge/router over ISDN lines, you can use a CLIP "dial string" to identify NETBuilder and other-vendor bridge/routers at the remote site to a central site bridge/router. The "dial string" is an ISDN phone number plus an optional subaddress. This number is configured with the -PORT CLIPList parameter.

If you are configuring a bridge/router port for multiple uses such as an ISDN dial-up path, a modem dial-up path, and a leased path, you should create a virtual port and identify the remote user using SCID.

You can configure a port to identify remote site devices with only SCID or with only CLIP. When you configure a virtual port for CLIP, you must use SCID to identify the remote site. The SCID identifier is ignored when the dynamic dial-up path binds to the virtual port.

If you have configured a port for SCID and CLIP, CLIP is used for incoming ISDN dial-up path calls. If the port is configured for something other than ISDN dial-up paths, SCID is used.

Configuring a Port to Use SCID

To configure a local bridge/router port to use SCID to identify the remote NETBuilder bridge/router, follow these steps:

- 1 Identify the central site to the remote sites using:

```
SETDefault -SYS SysCallerID = "<string>"
```

where "<string>" is a text string (within quotes) up to 31 characters long and unique to the system that identifies the central site (for example, a city name like "Santa Clara") to the remote sites.

- 2 Create a virtual port for the central site bridge/router that uses SCID to identify to the remote user, select SCID when creating the -PORT VirtualPort parameter, using:

```
ADD !<port> -PORT VirtualPort {<connectorID.channelID> {<FRDLCI> | <X.25  
DTE | SMDS}} | {SCID"<SysCallerID>"}
```

The SCID is a text string of characters within quotes that identifies the remote site, such as "San Diego," to the remote site.

To Configure a Port to Use CLIP

To configure a local bridge/router port to use CLIP to identify a remote NETBuilder or other bridge/router, follow these steps:

- 1 Select SCID for the -PORT VirtualPort parameter, using:

```
ADD !<port> -PORT VirtualPort {<connectorID.channelID> {<FRDLCI> | <X.25
DTE | SMDS}} | {SCID"<SysCallerID>"}
```

The SCID is a text string of characters within quotes that describes the remote site, such as "San Diego." This SCID is ignored when a dynamic path is binding to the virtual port configured for CLIP.

- 2 Add the dial string to the CLList database that identifies the remote site bridge/router to the central site virtual port using:

```
ADD !<port> -PORT CLList "dial string"
```

The dial string can be up to 50 alphanumeric characters in length and must be enclosed in quotes. The dial string is the remote site ISDN phone number. The phone number usually includes the dial prefix, country code, area code, and possibly a subaddress assigned to your ISDN interface. If you specify a subaddress, you must separate the phone number from the subaddress with a semicolon (;). With ISDN phone numbers, you can use hyphens (-) to separate the prefix, country code, and area code.

- 3 Set the central site bridge/router virtual port to check the identity of incoming ISDN calls with the CLList database by setting the -PORT DialRcvrState parameter to AnswerCLI using:

```
SETDefault !<port> -PORT DialRcvrState = NoAnswer | Answer | AnswerCLI
```

If the identity of the incoming call matches the dial string in the CLList database exactly, the dial-up path is bound to the port to make the connection.

Configuring PAP, CHAP and Standard Bundling

For information about configuring PAP, CHAP, and MS-CHAP authentication protocols and standard bundling using endpoint discriminators, see the Configuring Wide Area Networking Using PPP chapter.

Configuring Point to Point Tunneling Protocol

For information on how to configure a NETBuilder bridge/router as a tunnel terminator packet processor, how to configure a bridge/router as a tunnel initiator/terminator in a router-to-router configuration, and how to configure virtual leased lines with the Point-to-Point Tunneling Protocol (PPTP) see the Configuring L2Tunnel Connections chapter.

Configuring the Path Preference List

A path preference list reserves a path for use by a group of ports and sets the order of line use. Prioritization is accomplished by position in the path preference list. (Leased line paths cannot be included in the path preference list because bandwidth management cannot bring a leased line up or down; leased line paths are brought up when the port is enabled.) A path can be reserved by more than one port.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.

- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Enable your WAN communication resources according to “Configuring WAN Resources” earlier in this chapter.

Procedure

To configure the path preference list, follow these steps:

- 1 Paths added to the path preference list must be a dial-up line. To specify path 1 as dial up, enter:

```
SETDefault !1 -PATH LineType = Dialup
```

- 2 Unbind a path from its assigned port and add it to the dial pool by entering:

```
SETDefault !1 -PATH DialCONTROL = DYNAMIC
```

The WAN Extender virtual paths eligible for path preferences are already dynamic; the line type for virtual paths cannot be changed.

- 3 Enable changes to the path by entering:

```
SETDefault !1 -PATH CONTROL = Enabled
```

In this command, use the <path> syntax if you are specifying a DTE serial path and use the <connectorID.channelID> syntax if you are specifying an ISDN path.

- 4 Enable the port or virtual port to use dial pool resources and map the remote system caller ID to a specific port.

- a If you are using ports (as opposed to PPP virtual ports) with dynamic lines, use:

```
ADD !<port> -PORT PAtHs SCID"<SysCallerId>"
```

For example, to allow port 4 to use the dial pool for a path resource for outgoing calls and to map incoming calls with the caller ID of “London” to port 4, enter:

```
ADD !4 -PORT PAtHs SCID"London"
```

The string you enter for the caller ID is case-sensitive and can contain up to 31 characters.

- b If you are using PPP virtual ports, use:

```
ADD !<port> -PORT VirtualPort {SCID"<SysCallerId>"}
```

For example, to create PPP virtual port V3 and allow it to use the dial pool for its path resources for outgoing calls and to map incoming calls with the caller ID of “NewYork” to virtual port V3, enter:

```
ADD !V3 -PORT VirtualPort SCID"NewYork"
```

The caller ID string is case-sensitive and can contain up to 31 characters.

Unlike Frame Relay and X.25 virtual ports, which are always associated with a particular path, PPP virtual ports can potentially use any path in the dial pool.

- c Make sure each remote site has been configured with a unique caller ID using:

```
SETDefault -SYS SysCallerID = "<string>"
```

The SysCallerID string is limited to 31 characters. The string should be administratively assigned and be unique across the network.

- 5 Reserve the paths and define their priority using:

```
ADD !<port> -PORT PathPreference [<path>] [,...] [Pos = <1- number>]
```

For example, to specify paths 2.1 and 2.2 for use by port 5, enter:

```
ADD !5 -PORT PathPreference 2.1, 2.2
```

- 6 Enable the previous port changes by entering:

```
SETDefault !1 -PORT CONTROL = Enabled
```

- 7 Check the current configuration of the path preference list by entering:

```
SHOW -PATH DialPool
```

By default, the software adds dial paths to the end of the list if the position is not specified.

After paths are configured using the -PORT PathPreference parameter, no other ports can use the reserved paths except the designated ones. The software tries the preferred list of paths first before using path resources in the rest of the dial pool.



A dynamic path can appear in the path preference list for more than one port. A static path can only appear in the path preference list of the port to which it is bound.

You can append one or more dial-up paths to the end of the path preference list, insert one or more paths into a specific position in the list, or delete an existing path in the list. For more information, see the sections that follow.

Appending a Path

To append one or more dial paths to the end of the path preference list, use:

```
ADD !<port> -PORT PathPreference [<path>] [,...] [Pos = <1- number>]
```

For example, assume the path preference list for port 2 includes dial-up paths 3 and 2.1, and you want to append dial paths 5 and 6 to the end of the path preference list. Enter:

```
ADD !2 -PORT PathPreference 5, 6
```

After this command is executed, when port 2 needs a path resource, the software uses the preferred paths first. The order of their use is 3, 2.1, 5, and 6.

Adding a Path

To add one or more dial paths into a specific position in the path preference list, use the Pos (position) keyword with the desired position number. The paths are added into the list as follows:

- The software deletes any duplicate paths from the list.
- The software then adds the path list by inserting them starting at the specified position.

If you want to add more than one path, you must list the paths in the intended order.

For example, assume the path preference list for port 3 includes dial-up paths 3, 4, and 5, and you want to insert dial path 2.1 into position 2. Enter:

```
ADD !3 -PORT PathPreference 2.1 Pos = 2
```

After this command is executed, the path preference list has paths 3, 2.1, 4, and 5.

If you want to insert more than one dial path, you must list the paths in the intended order. For example, assume the path preference list for port 3 includes dial-up paths 3 and 5. To insert dial path 3 and 5 into position 2 and 3, enter:

```
ADD !3 -PORT PathPreference 3, 5 Pos = 2
```

After this command is executed, the path preference list is 2.1, 3, 5, and 4.

With this command, you can change the position of a path that already exists in the path preference list. For example, assume the path preference list for port 3 includes dial-up paths 4, 6, 7, and 5. To reposition path 6 into position 3, enter:

```
ADD !3 -PORT PathPreference 6 Pos = 3
```

The software inserts path 6 at position 3; path 7 that was originally in position 3 is now in position 2. If the position specified is larger than the existing list, the path is appended to the end of the list by default.

Deleting a Path To remove one or more dial-up paths from the path preference list, use:

```
DELEte !<port> -PORT PathPreference <path> [,...]
```

In this syntax, use <path> if you are specifying a DTE serial path. If you are specifying an ISDN path, substitute <connectorID.channelID> for <path>.

The paths to be deleted can be listed in any order. If you try to delete a path that does not exist in the list, an error message is displayed.

Configuring Manual Bandwidth Management Mode

With manual bandwidth management mode, you can control the connect sequence and bandwidth settings for a one-time call on a line. You use port-based dialing and the Dial command to manually dial on the specified port. The procedures in this section show how to configure manual bandwidth management mode and manually connect a line, and how to enable disaster recovery.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Enable your WAN communication resources according to “Configuring WAN Resources” earlier in this chapter.

Procedure

To place a call under manual bandwidth management mode, follow these steps:

- 1 Enable system bandwidth management mode using:

```
SETDefault !<port> -PORT DialInitState = ManualDial
```

- 2 Specify the amount of bandwidth that bandwidth management should bring up when this port is enabled using:

```
SET !<port> -PORT NORMAlBandwidth = <kbps>
```

- 3 Place the call using:

```
DIAL !<port> [-PORT] ["<dial-string>"]
```

This command accepts a static or dynamic port number. If you enter a telephone number in the optional dial string, it must also be listed in the dial number list; however, you can enter a port and telephone number to temporarily override

entries in the dial number list. See “Configuring the Dial List” earlier in this chapter for more information about setting up a dial list.

- 4 Add the telephone number to dial using:

```
ADD !<port> -PORT DialNoList "<phone no>" [Baud = <rate> (1.2-16000)] [Type
= Modem | Bri | Sw56 | WE | WEH0] | [Pos = <number>]
```

- 5 Disconnect the call using:

```
HangUp !<port> [-PORT]
```

HangUp brings down all dial path resources unless there are leased lines active.

Disaster Recovery Procedure

To configure disaster recovery using manual bandwidth management mode, follow these steps:

- 1 Enable manual bandwidth management by entering:

```
SETDefault !4 -PORT DialInitState = ManualDial
```

- 2 Enable disaster recovery on the port by entering:

```
SETDefault !4 -PORT DialCONTROL = DisasterRcvry
```

- 3 Specify the normal bandwidth for this port using:

```
SET !<port> -PORT NORMAlBandwidth = <kbps>
```

When you enable disaster recovery, you configure the software to bring up bandwidth to meet the target when a leased line goes down and bandwidth on the port falls below that specified with the -PORT NORMAlBandwidth parameter.

Verifying the Configuration

To verify the configuration, follow these steps:

- 1 Display dial-up configuration information using:

```
SHoW [!<port> | !*] -PORT DialCONFIg
```

This display shows the port state (up or down), port function (disaster recovery or BOD), the paths that are in use, the path state, and the dial string for active outgoing calls. The path in use can be a static path or a dynamic path from the dial pool. If the path is from the dial pool, the information is displayed similar to a static path.

The DialCONFIg parameter display identifies a WAN Extender virtual path as Dialup or Leased on the Dial Ctrl list if the port to which the virtual path is assigned is up; if the port is not up it displays a hyphen (-).

- 2 Display path configurations by entering:

```
SHoW -PATH CONFIguration
```

Verify that the paths are enabled and their status is up.

The CONFIguration parameter displays the Baud, Conn, and Line type for WAN Extender virtual paths only if the port to which they are connected is up. If the port is down, it displays a hyphen (-).

- 3 Display port configurations by entering:

```
SHoW -PORT CONFIguration
```

Verify that the ports are enabled and their status is up. Also make sure that dynamic ports are selecting path resources from the dial pool.

The SHoW -PORT CONFIguration parameter displays the SysCallerId (for example, SCID “Boston”) for dial-up and leased WAN Extender virtual paths if the port to

which they are connected is up. It also shows the aggregate bandwidth of the port. If the port is down, it displays a hyphen (-).

Troubleshooting the Configuration

To troubleshoot the configuration, follow these steps:

- 1 Display the status of lines under bandwidth management control using:

```
SHow [!<port> | !*] -PORT DialStatus
```

The display shows the state of the ports under bandwidth management. Displays include total port bandwidth utilization, and messages indicating congestion levels and the intentions of the bandwidth manager for allocating additional resources. If the port is to be used for an outgoing call, the dial string (phone number) is displayed.

The DialStatus parameter displays path number, B channel, and network port for WAN Extender virtual paths only if the port to which they are connected is up. If the port is down, it displays a hyphen (-).

- 2 Display a time-stamped dial history for the specified port or for all ports using:

```
SHow [!<port> | !*] -PORT DialHistory
```

Configuration Examples

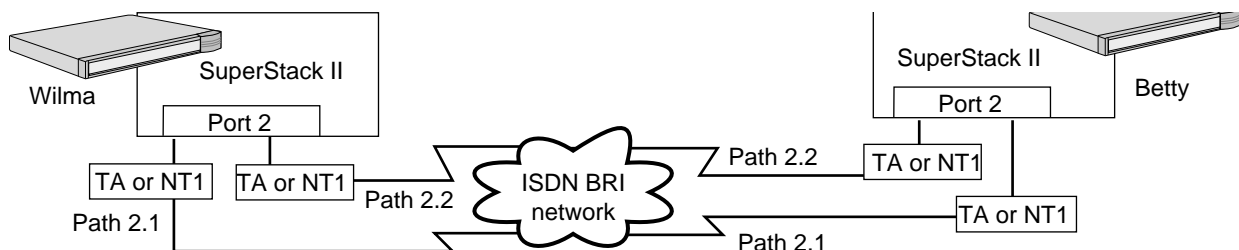
This section provides configuration examples that you can use to help you configure a wide-area network on your bridge/router using port bandwidth management.

Load Balancing over Multiple Dial-up Links

Load balancing equalizes traffic flow and makes sure that packets that may have been fragmented over the links arrive at their destination in the correct sequence. Load balancing is accomplished using the PPP Multilink Protocol (MLP) as described in RFC 1717. The -PPP MlpCONTROL parameter enables this protocol.

The example configuration depicted in Figure 333 uses system bandwidth management mode. Bandwidth is set to 64 kbps and BOD is triggered on when traffic exceeds 32 kbps; bandwidth management will add paths to increase bandwidth to the 128 kbps limit. The paths are taken down once traffic returns to 32 kbps or less for longer than 60 seconds.

Figure 333 PPP MLP Load Balancing



The following example configuration enables load balancing over two ISDN BRI channels. The parameters must be configured on both bridge/routers.

```
SETDefault !2 -Port name="ToBetty"
SETDefault !2.1 -Path LineType=Dialup
SETDefault !2.2 -Path LineType=Dialup
ADD !ToBetty -Port DialNoList "4085551212" t=bri baud=64
ADD !ToBetty -Port DialNoList "4085551313" t=bri baud=64
```

```

SETDefault !MLPPath1 -Path LocalDialNo="1234567"
SETDefault !MLPPath2 -Path LocalDialNo="2345678"
SETDefault !MLPPath1 -Path DialControl=(Static,UnRestricted)
SETDefault !MLPPath2 -Path DialControl=(Static,UnRestricted)
ADD !ToBetty -Port Path 2.1,2.2
SETDefault !2.1 -Path Control=Enabled
SETDefault !2.2 -Path Control=Enabled
SETDefault !ToBetty -Port DialInitState=DialOnDemand
SETDefault !ToBetty -Port NormalBandwidth=64
SETDefault !ToBetty -Port BODIncrLimit=64
SETDefault !ToBetty -Port BODThreshold=50
SETDefault !ToBetty -Port DialSamplePeriod=0,30
SETDefault !ToBetty -Port DialIdleTime=300
SETDefault !ToBetty -ppp MlpControl=Enabled
SETDefault !ToBetty -Port Control=Enabled

```

You can verify that both paths are up and that MLP is enabled by entering:

```

SHOW -PATH CONFIGURATION
SHOW -PPP STATUS
SHOW -PPP MlpSTATISTICS

```

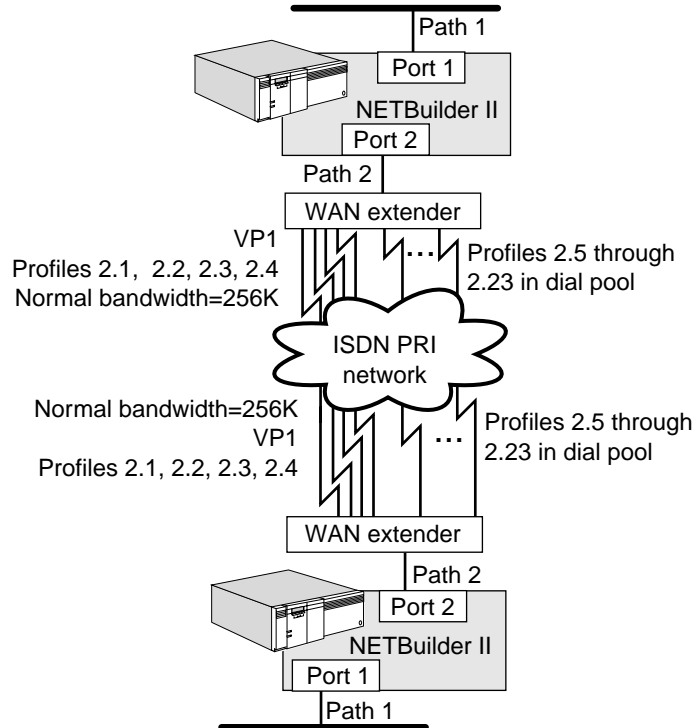
NETBuilder II WAN Extender Configuration Example

To configure the NETBuilder II bridge/router to use the WAN Extender virtual paths, you must set the owner of the port that corresponds to the bridge/router connection to WAN Extender. Set the baud rate on that port to 4096.

Virtual ports must be created to represent the logical attachment between the sites. Specify the mapping between the virtual port created as a representation of the logical attachment to the remote site and the WAN Extender profile that describes the basic physical connection in the dial number list. This must be done for each profile.

In Figure 334, two sites with NETBuilder II bridge/routers and WAN Extenders are to be configured with 256 kbps base bandwidth, and the remaining 19 B channels of the PRI line are to be made available for BOD.

Figure 334 WAN Extender Configuration Example



The first requirement is that 23 profiles be set up on the WAN Extender to enable calls between the two bridge/routers. These profiles describe virtual paths in a dial pool for the NETBuilder II bridge/router to use for port 2.

The following sample configuration shows how to set the baud rate, create the virtual ports, and configure the WAN Extender profiles.

```
SETDefault !2 -Port Owner = WanExtender
SETDefault !2 -Path BAud = 4096
ADD !V1 -Port VirtualPort WanExtender "SystemCallerID"
ADD !V1 -Port DialNoList "2 1" Type=WE
ADD !V1 -Port DialNoList "2 2" Type=WE
ADD !V1 -Port DialNoList "2 3" Type=WE
ADD !V1 -Port DialNoList "2 4" Type=WE
.
.
.
ADD !V1 -Port DialNoList "2 23"
```

The base bandwidth on the NETBuilder II bridge/routers are set to 256 kbps using the `NORMALbandwidth` parameter. The NETBuilder II bridge/router will dial up four B channels and keep them up at all times as a minimum bandwidth for the port. The upper bandwidth limit is set with the `BODIncrLimit` parameter. Setting this parameter to 1472 allows the full PRI line to be used if needed by the port. Setting the `BODIncrLimit` parameter to a level greater than zero enables BOD.

Two parameters control when the additional channels are to be dialed or hung up. The BODTHreshold parameter sets the trigger point stated as a percentage of the current active bandwidth. The DialSamplPeriod parameter sets how long the data rate has to remain over or under the BODTHreshold limit before a resource is dialed. In this case, the threshold is 75 percent and the sample period is 5 seconds.

These parameters configure a minimum 256 kbps pipe that is available all of the time. If the data rate through the pipe exceeds 192 kbps for 5 seconds, another 64 kbps channel will be added. If the data rate then exceeds 240 kbps for 5 seconds, another 64 kbps channel will be added and so on until all 23 channels are up for the port to use as long as the data rate exceeds 75 percent of the current bandwidth. If the data rate drops below 75 percent of the current bandwidth of 384 kbps (288 kbps) for 5 seconds, one of the B channels is dropped, and this continues as long as the data rate is less than 75 percent of the current bandwidth for 10-second intervals, down to the normal bandwidth of 256 kbps.

The following example configuration shows how to configure the bandwidth settings on the NETBuilder II bridge/routers.

```
SETDefault !V1 -Port DialInitState = DialOnDemand
SETDefault !V1 -Port NORMAlBandwidth = 256
SETDefault !V1 -Port BODIncrLimit = 1472
SETDefault !V1 -Port BODTHreshold = 75%
SETDefault !V1 -Port DialSamplPeriod = 5, 10
```

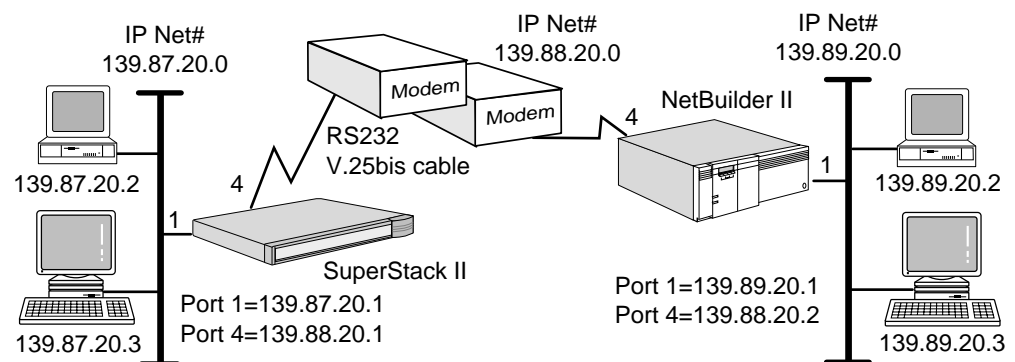
Routing Configurations over DOD Links

This section describes how to configure the network layer protocols supported with bandwidth management.

IP over a DOD Link

To configure IP over a DOD link, follow these steps. Figure 335 shows the IP network referenced in the following IP and RIPIP procedures.

Figure 335 IP Network Design Example



- 1 Configure the network address for port 1 and port 4 on the SuperStack II bridge/router by entering:

```
SETDefault !1 -IP NETAddr = 139.87.20.1
SETDefault !4 -IP NETAddr = 139.88.20.1
```

- 2 Configure the network address for port 1 and port 4 on the NETBuilder II bridge/router by entering:

```
SETDefault !1 -IP NETAddr = 139.89.20.1
SETDefault !4 -IP NETAddr = 139.88.20.2
```

- 3 Enable IP routing by entering the following command on the SuperStack II and NETBuilder II bridge/routers:

```
SETDefault -IP CONTROL = ROute
```

- 4 Add a static route on the SuperStack II bridge/router by entering:

```
ADD -IP ROUTe 139.89.0.0 139.88.20.2 1
```

- 5 Add a static route on the NETBuilder II system by entering:

```
ADD -IP ROUTe 139.87.0.0 139.88.20.1 1
```

- 6 Check transport and network layer status by entering:

```
SHow -IP NETaddr
```

RIP over a DOD Link To configure DOD on a RIP network, follow these steps:

- 1 On the SuperStack II bridge/router, configure the LAN ports to send and receive update packets by entering:

```
SETDefault !1 -RIP CONTROL = (TALK, Listen)
```

- 2 On the SuperStack II bridge/router, configure the WAN ports to not run the RIP Protocol, but instead take advantage of the static routes you set up using IP by entering:

```
SETDefault !4 -RIP CONTROL = (NoTALK, NoListen)
```

For more information on static routes, see "IP over a DOD Link" earlier in this chapter.

- 3 On the NETBuilder II system, configure the LAN ports to send and receive update packets by entering:

```
SETDefault !1 -RIP CONTROL = (TALK, Listen)
```

- 4 On the NETBuilder II system, configure the WAN ports to not TALK and not Listen by entering:

```
SETDefault !4 -RIP CONTROL = (NoTALK, NoListen)
```

- 5 Advertise static policies by entering the following command on the SuperStack II and NETBuilder II bridge/routers:

```
ADD !1 -RIP StaticPolicy All
```

TCP for SNA Traffic over a DOD Link

To use the recommended TCP protocol settings to enable Systems Network Architecture (SNA) traffic, including data link switching (DLSw), to be sent over an ISDN DOD link, follow these steps:

- 1 On the NETBuilder II or SuperStack II bridge/routers, disable TCP keepalive packets by entering:

```
SETDefault -TCP CONTROL = NoKeepAlive
```

- 2 Set the TCP retransmit limit to either 3 or 4 using:

```
SETDefault -TCP RetransmitLimit = <retrys> (0-128)
```

The limit of either 3 or 4 is specifically recommended for SNA configurations over DOD.

- 3 Set the time in seconds before a DOD line is disconnected using:

```
SETDefault !<port> -PORT DialIdleTime = <seconds> (0-3600)
```

The default is three minutes (180 seconds).

- 4 Set the DOD retry count on the port to 9 using:

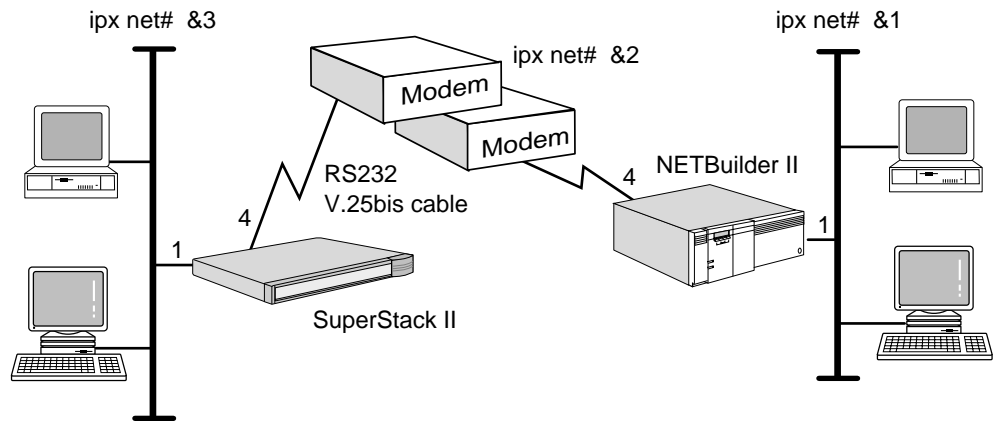
```
SETDefault !<port> DialRetryCount = 9
```
- 5 Set the DOD retry timer on the port to 15 seconds using:

```
SETDefault !<port> -PORT DialRetryTime = 15
```

IPX with Incremental Broadcasts over a DOD Link

Figure 336 shows the IPX network referenced in the following procedure. See the Configuring IPX Routing chapter for more information on IPX routing.

Figure 336 IPX Network Design Example



To configure DOD with incremental broadcasts on an IPX network, follow these steps:

- 1 Set the network number of port 1 and port 4 on the SuperStack II bridge/router by entering:

```
SETDefault !1 -IPX NETnumber = &3  
SETDefault !4 -IPX NETnumber = &2
```
- 2 Set the network number for port 1 and port 4 on the NETBuilder II bridge/router by entering:

```
SETDefault !1 -IPX NETnumber = &1  
SETDefault !4 -IPX NETnumber = &2
```
- 3 Enable IPX routing on both routers by entering:

```
SETDefault !1 -IPX CONTROL = ROute  
SETDefault !4 -IPX CONTROL = ROute
```
- 4 Disable WAN broadcasts on LAN ports by entering:

```
SETDefault !1 -IPX CONTROL = NoIpxWan
```
- 5 Change NRIP updates on port 4 of both routers from periodic broadcast to incremental broadcast by entering:

```
SETDefault !4 -NRIP CONTROL = NoPeriodic
```
- 6 Change SAP updates on port 4 of both routers from periodic broadcast to incremental broadcast by entering:

```
SETDefault !4 -SAP CONTROL = NoPeriodic
```

See the Configuring IPX Routing chapter for more information on RIP/SAP updates.

IPX Protocol in a Boundary Routing Environment over a DOD Link

To help you configure DOD with the IPX Protocol in a Boundary Routing environment, two configuration examples are provided.

Example 1

Port 3 of the NETBuilder II bridge/router at the central site is linked with port 3 of the SuperStack II boundary router peripheral node. NetWare clients but no NetWare servers exist on the peripheral network. NetWare clients attach across the DOD link back to the NetWare servers on the central site.

To configure IPX in a Boundary Routing environment, follow these steps:

- 1 Enable Boundary Routing on port 3 of the NETBuilder II bridge/router by entering:

```
SETDefault !3 -BCN CONTROL = Enabled
```

- 2 Enable IPX routing on the NETBuilder II bridge/router by entering:

```
SETDefault !3 -IPX CONTROL = ROute
```

- 3 Assign the IPX network number on port 3 of the NETBuilder II bridge/router by entering:

```
SETDefault !3 -IPX NETnumber = &2001
```

- 4 Use incremental NRIP and SAP to reduce broadcast traffic on the DOD link by setting port 3 on the NETBuilder II bridge/router by entering:

```
SETDefault !3 -NRIP CONTROL = (NoPEriodic)
```

```
SETDefault !3 -SAP CONTROL = (NoPEriodic)
```

- 5 Verify that spoofing of NetWare Core Protocol (NCP) keep alive packets using the WatchDog mechanism is enabled on the NETBuilder II bridge/router by entering:

```
SHow !3 -IPX SPOofCONTROL
```

If spoofing has been disabled (NoNcpWatchDog), enable it by entering:

```
SETDefault !3 -IPX SPOofCONTROL = NcpWatchDog
```

See the Configuring IPX Routing chapter for more information about NCP spoofing.

In this example, because there are no NetWare servers at the remote site, NetWare clients log on the servers at the central site. Consequently, there is periodic traffic of NCP keep alive packets between the servers and clients in order to maintain these NCP connections.

Example 2

Port 3 of the NETBuilder II bridge/router at the central site is linked with port 3 of the SuperStack II boundary router peripheral node. NetWare servers and NetWare clients exist on the peripheral network. On the peripheral network, NetWare clients log on to the remote servers as their primary servers and only attach across the DOD link to the central site servers periodically whenever their application needs dictate, for example, reading electronic mail.

To specify this configuration, follow these steps:

- 1 Enable Boundary Routing and smart filtering on port 3 of the NETBuilder II bridge/router by entering:

```
SETDefault !3 -BCN CONTROL = (Enabled, SmartFiltering)
```


- 2 Enable IPX routing on the NETBuilder II bridge/router by entering:

```
SETDefault !3 -IPX CONTROL = ROute
```

- 3 Assign the IPX network number on port 3 of the NETBuilder II bridge/router by entering:

```
SETDefault !3 -IPX NETnumber = &2001
```

- 4 Use periodic NRIP and SAP in conjunction with smart filtering by setting port 3 on the NETBuilder II bridge/router by entering:

```
SETDefault !3 -NRIP CONTROL = (Talk, Listen, PEriodic)
```

```
SETDefault !3 -SAP CONTROL = (Talk, Listen, PEriodic)
```

In this example, with servers on the remote sites, NetWare clients should log on to these servers as their main servers, and occasionally log on (attach) to the servers at the central site. To reduce the NCP keepalive packets across the DOD link, you can set user guidelines to request that users only maintain their login to a central site server when their application needs it.

Summary of Bandwidth Manager Commands and Parameters

Table 86 summarizes the commands and parameters that are used with the port bandwidth management.

Table 86 Bandwidth Management Tasks and Commands

| Task | Command or Parameter | Description | Applies to |
|------------------|----------------------|--|--|
| Configure a path | -PATH LineType | Sets the type of line being used. | DTE, ISDN, WE |
| | -PATH CONNector | Specifies the type of connector for a serial interface. | DTE |
| | -PATH DialMode | Configures V.25 bis standard dialing or dialing from a DTR modem. | DTE |
| | -PATH ExDevType | Specifies the device type attached to the DTE path. | DTE, ISDN |
| | -PATH SwitchType | Specifies the type of ISDN switch to which an ISDN path is connected. | ISDN |
| | -PATH LocalDialNo | Associates a phone number to your ISDN path. | ISDN on model 42x and 52x SuperStack II and ISDN on model 14x OfficeConnect NETBuilder |
| | -PATH LocalSubAddr | Specifies a subaddress to the phone number you specified for your ISDN path. | ISDN on model 42x and 52x SuperStack II and ISDN on model 14x OfficeConnect NETBuilder |

Table 86 Bandwidth Management Tasks and Commands (continued)

| Task | Command or Parameter | Description | Applies to |
|--|---------------------------|---|--|
| | -PATH SPIDdn1 and SPIDdn2 | Specifies the Service Profile Identifiers (SPIDs) and directory numbers (DNs) for North American BRI ISDN dial-up modes. | ISDN on model 42x and 52x SuperStack II and ISDN on model 14x OfficeConnect NETBuilder |
| | -PATH RateAdaption | Specifies a method that determines the data rate to be used on an ISDN path. | ISDN on model 42x and 52x SuperStack II and ISDN on model 14x OfficeConnect NETBuilder |
| Enable manual bandwidth management mode (manual dial mode) | -PORT DialInitState | ManualDial option allows you to manually dial calls. You specify bandwidth settings rather than let bandwidth management monitor the line and adjust settings as needed. | DTE, ISDN, WE |
| Connect | Dial | Manually connects a dial-up path or port. | DTE, ISDN, WE |
| | -PORT CLList | Adds (or deletes) a "dial string" (the ISDN phone number and subaddress) to a list of ISDN numbers to be used by the called party to map the incoming call to the appropriate port and to bind an ISDN dynamic dial-up path to the port to complete the call. This parameter is also used to screen out any calls that do not have a match in the CLList database.

The CLList entries can take effect only if -PORT DialRcvrState has been set to AnswerCLI. See the -PORT DialRcvrState parameter in this section for more details.

The binding of a path to a port with a CLI number supersedes and ignores the binding between path and port set up by a system caller ID (SCID) number. | ISDN |
| | -PORT DialNoList | Adds, deletes, edits, and displays a list of phone numbers with their associated attributes (baud rate, phone number, and position in the list) for a bridge/router port. The bridge/router port tries to match an incoming call with this list of phone numbers. If there is a match, the port tries to find a path to make the connection. | DTE, ISDN, WE |
| | -PORT AutoDial | Connects the dial-up path assigned to a port as soon as the path is enabled. | DTE, ISDN, WE |
| Disconnect | HangUp | Manually disconnects a dial-up path or port. | DTE, ISDN, WE |
| | -PATH DialCarrierTime | Defines the number of seconds the system must wait for carrier signals on the line that has connected. | DTE, ISDN, WE |
| | -PORT DialIdleTime | Sets the idle timer in seconds for a dial-up line before the line is disconnected if it is not in use. | DTE, ISDN, WE |

Table 86 Bandwidth Management Tasks and Commands (continued)

| Task | Command or Parameter | Description | Applies to |
|--|-----------------------|---|---------------|
| Retry a dial-up connection | -PORT DialRetryCount | Specifies the number of times to retry the call if the call attempt fails. | DTE, ISDN, WE |
| | -PORT DialRetryTime | Sets the initial value in seconds to wait before attempting to reconnect after a connection has failed because the carrier was not detected or for any other reason that the path did not come up. | DTE, ISDN, WE |
| Configure port attributes for answer-only line | -PORT DialRcvrState | Sets the call receiver dial control state. Also, selecting AnswerCLI specifies that the bridge/router will try to match an incoming call to the port specified and to bind an ISDN dynamic dial-up path to that port if the ISDN number (and subaddress) matches exactly the ISDN number in the CLList database. See the -PORT CLList parameter for more details. | DTE, ISDN, WE |
| | -PORT DialInitState | NoDialOut option prevents outgoing calls when -PORT DialRcvrState is set to Answer. | DTE, ISDN, WE |
| Enable system bandwidth management mode (dial-on-demand) | -PORT DialOnDemand | DialOnDemand option enables system bandwidth management for bandwidth-on-demand and dial-on-demand modes, and monitors the line; additional lines are brought up or down based on traffic demand. | DTE, ISDN, WE |
| Configure for bandwidth-on-demand | -PATH DialCONTRol | Assigns the dial path unrestricted use as an additional resource for adding bandwidth. | DTE, ISDN |
| | -PORT NORMAlBandwidth | Specifies the port bandwidth setting. | DTE, ISDN, WE |
| | -PORT BODThresholD | Configures the threshold that triggers the BOD line up and down. | DTE, ISDN, WE |
| | -PORT BODIncrLimit | Configures the maximum incremental bandwidth that can be allocated using BOD. | DTE, ISDN, WE |
| | -PORT DialSamplPeriod | Sets the time to sample threshold conditions before taking an action to bring a path up or down, based on transmit traffic load for BOD. | DTE, ISDN, WE |
| Configure disaster recovery | -PORT DialCONTRol | Restricts the dial path for use as disaster recovery only. | DTE, ISDN, WE |
| Configure a port to use dynamic dial path resources | -PORT PAtHs | Assigns dial pool resources to a specified port and identifies the remote system caller ID. For dial-up lines of any kind, the remote caller ID is a text string (like a city name). | DTE, ISDN, WE |
| | -PORT VirtualPort | Creates a virtual port that uses path resources from the dial pool and identifies the path and circuit types attached to the port. It also identifies the remote site associated with the virtual port. | DTE, ISDN, WE |
| | | To configure a bridge/router port for multiple uses (dial-ups of any kind), identify the remote site with a system caller ID (SCID) text string. The SCID for remote sites can be a telephone number or a text string. | |

Table 86 Bandwidth Management Tasks and Commands (continued)

| Task | Command or Parameter | Description | Applies to |
|--|---|---|----------------------------|
| | | When you configure ISDN dial-up lines, you can also identify remote sites with a Calling Line Identification Presentation (CLIP) dial string, which is the remote site ISDN telephone number. | |
| | | Remote sites identified with SCID can only connect 3Com remote sites to the central site, while remote sites identified with CLIP can connect 3Com and other-vendor bridge/routers at the remote sites with the central site. | |
| Configure path attributes for a dial-up path | -PATH DialCONTRol

-PORT PathPreference | Sets path attributes for static and dynamic dial-up paths.

Configures the dial path usage preference. | DTE, ISDN

DTE, ISDN |
| Display port and path status | -PORT DialSTatus | Provides the status of the dial-up service and the state of bandwidth management for the specified dial ports. | DTE, ISDN, WE |
| Display port dial history | -PORT DialHistory | Provides the dial history for the specified port. | DTE, ISDN, WE |
| Display the dial pool status | -PATH DialPool | Provides the status and configuration of the paths in the dial pool. | DTE, ISDN, WE |

Bandwidth Management Concepts

This section explains the concept of bandwidth management and lists the resources the bandwidth management feature manages. Before proceeding, you need to be familiar with the concepts of ports and paths as described in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter. A glossary of terms used in this chapter is provided at the end of the chapter.

Virtual Pipe Port *bandwidth management* is a process that applies static bandwidth, dynamic bandwidth, or a combination of these to provide a port with the bandwidth it needs to meet current requirements. At each port, a set of serial path resources are configured to provide a bandwidth bundle called a *virtual pipe*.

Bandwidth Static bandwidth is provided by a configuration of one or more leased lines or dial paths to a port. The static resources are dedicated to a single port. Leased lines can provide continuous dedicated bandwidth to the port. Static dial paths can also provide incremental bandwidth, or bandwidth that becomes available only when a decision is made to dial them up.

Dynamic bandwidth is provided by dial-up line paths, which are allocated from a *dial pool*. Incremental bandwidth is provided by dial paths. The port can use either analog or digital lines that are allocated to it, and additional dial path resources can be added incrementally.

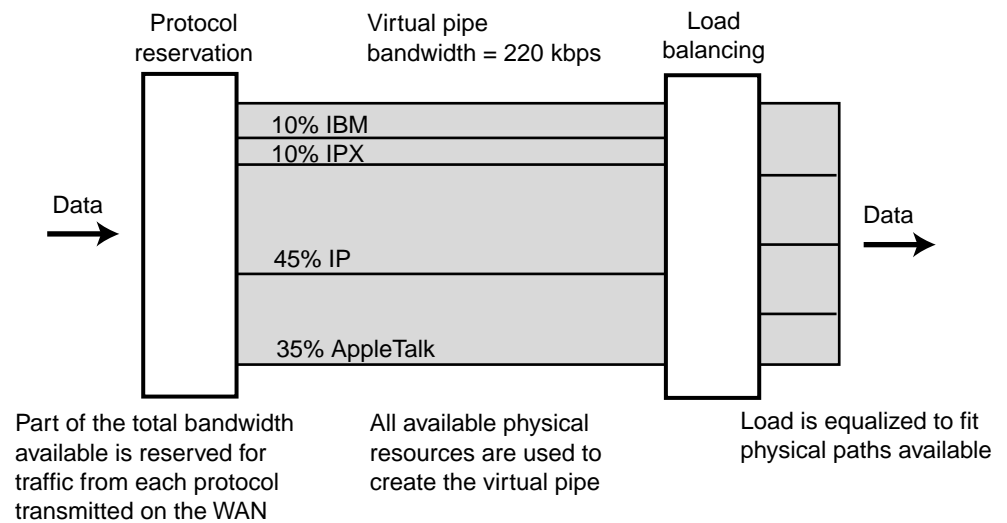
Bandwidth management operates on a port-by-port basis. It monitors line use based on rate of traffic and increases or decreases bandwidth based on limits that you specify. Network protocols that use the port are unaware of the underlying

physical links, which bandwidth management bundles together into the virtual pipe to meet the port bandwidth requirements.

Bandwidth Aggregation

The main function of bandwidth management is to determine the aggregate bandwidth that will be provided to the set of protocols passing through the port. However, a WAN operates most efficiently when it can allow for variations in the type and amount of traffic passing through it. In addition to bandwidth management, the software provides the protocol reservation feature, which allocates portions of the virtual pipe to specified traffic such as the Internet Protocol (IP) or AppleTalk. As traffic passes through the pipe, the Point-to-Point (PPP) Multilink Protocol (MLP) can also be enabled to distribute packets more evenly over the virtual pipe. Figure 337 illustrates these concepts.

Figure 337 Use of Resources through the Virtual Pipe



You reserve bandwidth for the protocols traversing the WAN using the `-PORT ADD ProtocolRsrv` parameter; see the *Configuring Mnemonic Filtering* chapter for further explanation.

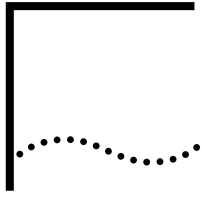
Load balancing equalizes traffic flow and makes sure that packets that may have been fragmented over the links arrive at their destination in the correct sequence. Load balancing is accomplished using the PPP Multilink Protocol as described in RFC 1717 and is enabled using the `-PPP MlpCONTROL` parameter.

Bandwidth Management Terms

The following terms are used in this chapter to explain concepts such as dial pools and the ports that can use them, and the bandwidth management strategies.

| | |
|----------------------|--|
| bandwidth management | A process that applies static bandwidth, dynamic bandwidth, or a combination of these to provide a port with the bandwidth it needs to meet current requirements. See also <i>virtual pipe</i> . |
| dial pool | The pool of dial paths that can be dynamically bound to any properly configured port. |

| | |
|-----------------------------|---|
| disaster recovery threshold | The minimum of the normal bandwidth threshold and the total amount of configured leased line bandwidth that is assigned to the port. See also <i>normal bandwidth threshold</i> . |
| dynamic binding | The association of a path in the dial pool to a port when it is needed. |
| dynamic path | <p>A path that can be used by more than one port. You create a dynamic path by unbinding it from its port. A dynamic path is stored in the dial pool. Characteristics of dynamic paths are as follows:</p> <ul style="list-style-type: none">■ Initially, dynamic paths are not bound to any port, but dynamically bind to make an outgoing call. To receive an incoming call, a dynamic path receives the call while still in the dial pool, and then is bound to a port.■ Dynamic paths can bind to different ports without user action, but only one port can bind at a time.■ Once the path becomes inactive, it unbinds from the port and becomes available for other ports. |
| normal bandwidth threshold | Bandwidth threshold defined by the -PORT NORMAlBandwidth, BODTHreshold, BODIncrLimit, and DialSamplPeriod parameters. |
| port | <p>A port is a logical interface used by the software to represent a connection to a network.</p> <p>When the DOD path is up, the bridge/router routes the packets as expected in the normal NCP and SPX1 connection processes.</p> |
| static binding | The association or binding of a path to a port as defined at system initialization time or by user configuration. |
| static path | A path is assigned (bound) to one port and can be used only by one port; a static path cannot be shared. By default, all paths are static at system initialization time. |
| virtual path | <p>A path used by the NETBuilder II bridge/router to represent multiple logical paths multiplexed over a single interface. ISDN B, ISDN PRI, and DSO channels delivered by channelized T1/E1 or switched-56 are presented as distinct virtual paths. A virtual path can be used as a static or dynamic resource.</p> |
| virtual pipe | A term that describes a port of variable bandwidth. |
| virtual port | A port that is not associated with a physical interface. Virtual ports allow configuration of multiple destinations through a single interface. |



CONFIGURING WIDE AREA NETWORKING USING FRAME RELAY

This chapter describes how to configure your bridge/router to establish serial line connectivity through Frame Relay. It also describes how this wide area protocol works and gives guidelines for operating and managing it.

The Asynchronous Transfer Mode data exchange interface (ATM DXI) on the bridge/router operates as part of the Frame Relay service. Most of the procedures in this chapter for configuring Frame Relay can also be used to configure the ATM DXI.



For conceptual information, see “How Frame Relay Works” later in this chapter.

Setting Up the Frame Relay Service

This section describes how to configure your bridge/router to transmit and receive data over a Frame Relay interface.

You must follow the steps in this section whether you are configuring for bridging or for routing. After you have completed these steps, proceed to “Setting Up Basic Bridging over Frame Relay” for bridging configuration information or to “Setting Up Basic Routing over Frame Relay” for routing configuration information.

For detailed descriptions of all commands, see *Reference for Enterprise OS Software*.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your wide area bridge/router ports and paths.
- Determine if your Frame Relay network supports the Local Management Interface (LMI) Protocol. For information about LMI, see “Local Management Interface Protocol” later in this chapter.
- Determine if you have a partially meshed or nonmeshed topology. If you do and you plan to enable the Internet Protocol (IP), the Routing Information Protocol (RIP), the Internet Packet Exchange (IPX), or AppleTalk routing, make certain that the next-hop split horizon feature is enabled. If you have a partially meshed or nonmeshed topology and you plan to enable Open Shortest Path First (OSPF), make sure that you set -OSPF CONTROL to NonMesh to enable the point-to-multipoint interface. If you plan to enable bridging, Xerox Network Systems (XNS), VINES, or DECnet IV routing, make sure that you have created virtual ports for each remote network that is attached to a Frame Relay cloud. For information on meshed, partially meshed, and nonmeshed topologies, next-hop split horizon, and virtual ports, see “How Frame Relay Works” later in this chapter. For instructions on setting up virtual ports, see the Configuring Advanced Ports and Paths chapter.

Procedure To transmit and receive data over a Frame Relay network, follow these steps:

- 1 Enable the Frame Relay service by setting the owner of the serial interface to Frame Relay using:

```
SETDefault !<port> -PORT OWNeR = FrameRelay
```



If PORT OWNeR is set to Auto, Frame Relay is detected and configured automatically and this step may not be necessary.

For networks running RIP with the port up and the -RIPiP CONTrol parameter set to TAIK, the CONTrol parameter DynamicNbr option is automatically enabled. With the DynamicNbr enabled, neighbors are automatically added. If DynamicNbr is not enabled, neighbors must be added manually.

- 2 If your Frame Relay network supports the LMI Protocol, make sure that the appropriate LMI Protocol is enabled. If your Frame Relay network does not support the LMI Protocol, disable this protocol.

The Enterprise OS software includes four types of LMI: Consortium LMI, Annex-D LMI, NTT LMI, and ITU LMI. Configure the software with the type of LMI that the switching equipment supports. Configure Consortium LMI by specifying LMI; configure Annex-D LMI by specifying ANSiLMI; configure NTT LMI by specifying the value NTTLMI; configure ITU LMI by specifying the value ITULMI.

If the port is configured for auto detect, the type of LMI is determined dynamically. To manually enable the specific LMI or to completely disable the LMI Protocol, use:

```
SETDefault !<port> -FR CONTrol = [NoLMI | LMI | ANSiLMI | NTTLMI | ITULMI]
```

Configuring Congestion Control

You can configure congestion control for individual virtual circuits on a logical or virtual port you are configuring for Frame Relay. There is a procedure to configure congestion control for NETBuilder bridge/router ports configured for the NTTLMI protocol, and there is another procedure to configure congestion control for ports configured to use LMI protocols other than NTTLMI. For more information, see “Frame Relay Congestion Control” later in this chapter .

For NTTLMI Protocol Users

To configure congestion control for NETBuilder bridge/router ports configured for NTTLMI, use:

```
SETDefault !<port> -FR DLCIR = <vcid> <cir>
```

where the Frame Relay <vcid> (virtual circuit identifier) value for a permanent virtual circuit (PVC) is the data link circuit identifier (DLCI) assigned by your Frame Relay service provider from a range of 16 through 991. The <vcid> value for a switched virtual circuit (SVC) is a unique virtual circuit identifier number that you assign to an SVC from the 16 through 991 DLCI range of numbers that has not been assigned by your service provider for a PVC. For more information on the -FR DLCIR parameter, see the FR Service Parameters chapter in *Reference for Enterprise OS Software*.

<cir> specifies the rate of NETBuilder bridge/router data (in kilobits per second) that the Frame Relay network commits to transfer under normal conditions.



CAUTION: *Failure to specify <cir> values for the DLCIs causes unpredictable results.*

For Other LMI Protocol Users

To configure congestion control for NETBuilder bridge/router ports configured to use an LMI protocol other than NTTLMI, follow these steps:



See the *FR Service Parameters chapter in Reference for Enterprise OS Software for more information on the parameters described in these steps.*

- 1 To activate congestion control for a particular port and individual <vcid> connection, and to specify how many consecutive BECN=1 frames (<step>) Frame Relay sends to the bridge/router port before the maximum throughput rate is reduced to a level below cir, use:

```
SETDefault !<port> -FR CongestControl = <vcid> [YES | NO] <step> (1-999)
```

- 2 To specify the throughput parameters for data coming in and going out a specified virtual circuit (<vcid>) connection on a specified port, use:

```
SETDefault !<port> -FR CIRbothdir = <vcid> <cir> <mincir> <Bc> <Be>
```

where:

- <vcid> identifies the identification of the Frame Relay virtual circuit used to establish the connection between the remote user and the local port.
- <cir> specifies the rate of NETBuilder bridge/router data (in Kilobits per second) that the Frame Relay network commits to transfer under normal conditions.
- <mincir> specifies the minimum rate of NETBuilder bridge/router data throughput (in Kilobits per second) that the calling user is committed to accept for the call. Normally <mincir> is specified for SVCs and not PVCs. The <mincir> value will be specified for PVCs only if the PVC <cir> is set to zero and the user wants to use congestion control.
- <Bc> specifies the maximum of NETBuilder bridge/router data bits (in Kilobits) that the network commits to transfer under normal conditions during the time interval (Tc) measured in seconds.
- <Be> specifies the maximum of uncommitted NETBuilder data bits (in Kilobits) in excess of Bc that the network attempts to deliver during the time interval (Tc) measured in seconds. If <Be> is nonzero, the NETBuilder bridge/router will not transmit more than <Be> number of data bits over the Tc time interval.



When configuring a PVC, the values entered for <cir> , <Bc>, and <Be> must match the values for these variables provided by your service provider.

- 3 To enable (disable is the default) Frame Relay to send messages when the network is congested and uncongested to the LLC2 layer use:

```
SETDefault -LLC2 FRCongestCont = ([Enable | Disable])
```



Setting Frame Relay Congestion Control for a port that is configured for LLC2 and SNA over Frame Relay increases overall network performance and reliability. On very large networks, congestion control for a port configured for LLC2 and SNA over Frame Relay can lower the performance of SNA because of the messages sent by Frame Relay when the network is congested or uncongested. Poor response time is an indicator that the SNA performance is being affected by Frame Relay Congestion Control.

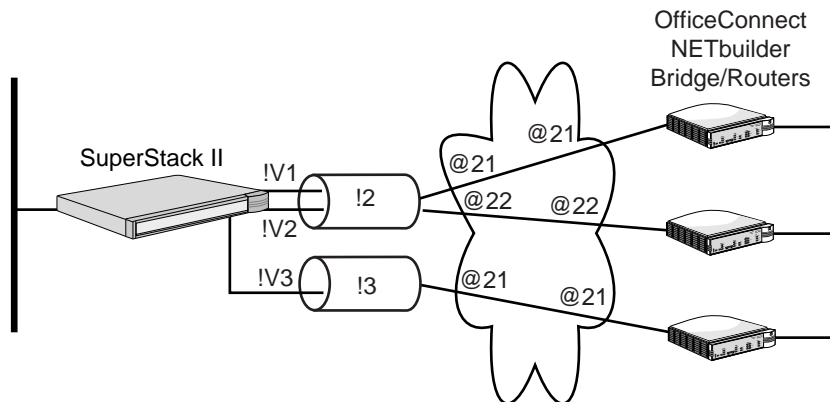
See "FRCongestCont" in the LLC2 Service Parameters chapter in *Reference for Enterprise OS Software* for details about the -LLC2 FRCongestCont parameter.

Example

This example describes how to configure the logical ports of the SuperStack II NETBuilder bridge/router shown in Figure 338 for Frame Relay Congestion Control. The following values are used in the configuration:

- Virtual port 1 on physical path 2 over DLCI (vcid) 21 with a <cir> of 32, <mincir> of 32, <Bc> of 32, and a <Be> of 32.
- Virtual port 2 on physical path 2 over DLCI (vcid) 22 with a <cir> of 16, <mincir> of 16, <Bc> of 16, and a <Be> of 16.
- Virtual port 3 on physical path 3 over DLCI (vcid) 22 with a <cir> of 16, <mincir> of 16, <Bc> of 16, and a <Be> of 32.
- All ports are set to a <step> value of 4.

Figure 338 Congestion Control Configuration Example



- 1 To activate congestion control for all virtual circuits, and to specify 4 consecutive BECN=1 frames (<step>) Frame Relay sends to the bridge/router port to notify it of congestion on the specific virtual circuit before the maximum throughput traffic is reduced below cir, enter:

```
SETDefault !V1 -FR CongestControl 21 Yes 4
SETDefault !V2 -FR CongestControl 22 Yes 4
SETDefault !V3 -FR CongestControl 21 Yes 4
```

- 2 To specify the <cir>, <mincir>, <Bc>, and <Be> values for all the logical ports, enter:

```
SETDefault !V1 -FR CIRbothdir 21 32 32 32 32
SETDefault !V2 -FR CIRbothdir 22 16 16 16 16
SETDefault !V3 -FR CIRbothdir 21 64 64 64 0
```

Configuring PVCs and SVCs

You can use either PVCs or SVCs to connect remote users through the Frame Relay network to your bridge/router port. PVCs are identified by Frame Relay virtual circuit IDs (<vcid>), which are actually DLCI numbers assigned by your service provider to make the connection. The range of DLCIs from which the service provider can assign is determined by the LMI protocol that you configured for your NETBuilder bridge/router port.

SVCs are identified by Frame Relay virtual circuit IDs, which are numbers that match DLCI numbers available from the range of DLCIs that have not been assigned by the service provider for PVC connections.

Unlike PVCs, which use the DLCIs to make their connections, SVCs must be configured for the following addresses (telephone numbers) to establish their connections:

- The customer premises equipment (CPE) link address, which is the telephone number assigned by your network service provider for your local bridge/router port. This address is used by all SVCs configured for this port.
- The destination address (telephone number) of the remote user associated with the local virtual circuit ID
- The address of the local SVC, which is the telephone number associated with the local virtual circuit ID (the DLCI you assigned it). The CPE link address is used to identify the local virtual circuit if the address of the local SVC is not set

SVCs can be dynamic connections that go down after the data transfer has stopped for the time interval specified with the `-FR SvcIdleTimer` parameter or they can be configured as static connections that remain up after the data transfer has stopped, like PVCs.

SVC Configuration Example To configure SVC connections for your NETBuilder bridge/router port, follow these steps:



See the FR Service Parameters chapter in *Reference for Enterprise OS Software* for more information on the parameters described in these steps.

- 1 Identify the local customer premises equipment (CPE) address (telephone number) assigned by your Frame Relay network service provider for your NETBuilder bridge/router, which identifies the local bridge/router port to the network, using:

```
SETDefault !<port> -FR LinkAddress = [<E.164 address> (1-15 digits) |
<X.121 address> (1-15 digits)]
```

- 2 Identify the SVC destination address (telephone number) of the remote user associated with the local virtual circuit ID (<vcid>) using:

```
ADD !<port> -FR SvcDestAddress = <vcid> [<E.164 address> (1-15 digits) |
<X.121 address> (1-15 digits)]
```

- 3 Identify the local SVC telephone number associated with a specified virtual circuit ID (<vcid>) to the remote user (optional) using:

```
SETDefault !<port> -FR SvcLocalAddress = <vcid> [<E.164 address> (1-15
digits) | <X.121 address> (1-15 digits)]
```

where <vcid> specifies a virtual circuit identifier number from the range of DLCI numbers available and that have not been assigned by your service provider for a PVC connection. The local address configured for this <vcid> number is used to identify the local virtual circuit used to establish the connection between the remote user and the bridge/router port. If the address of the local SVC is not set, the CPE link address is used to identify the virtual circuit.

- 4 An SVC connection is automatically established depending on whether the port is busy with another caller on another <vcid> assigned to the port.

To manually activate an SVC connection that ensures a connection for your <vcid> to that port (optional), use:

```
ADD !<port> -FR SvcConnection <vcid>
```

To manually disconnect an SVC connection for your <vcid> to that port (optional), use:

```
DElete !<port> -FR SvcConnection <vcid>
```

- 5 To set your SvcIdleTimer parameter for how long an SVC will remain idle before it shuts down (optional), use (the default is 80 seconds):

```
SETDefault!<port> -FR SvcIdleTimer <vcid> [None | (0-3600 seconds)]
```

- 6 To either enable or disable SVC operation on the specified port (optional), use:

```
SETDefault !<port> -FR SvcMode = [0 = disabled | 1 = enabled]
```

The default setting for this parameter is 0 (disabled).

Verifying the Configuration

To verify the Frame Relay configuration, enter:

```
SHow -FR CONFIguration
```

The router displays current Frame Relay configuration information.

Setting Up Basic Bridging over Frame Relay

This section describes how to configure transparent and source route bridging over Frame Relay.

Configuring Transparent Bridging

This section provides information for configuring transparent bridging over Frame Relay.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in *Using Enterprise OS Software*.
- Set up the Frame Relay service as described in “Setting Up the Frame Relay Service” earlier in this chapter.
- If your Frame Relay network does not support the LMI Protocol and you want to bridge to specific DLCI neighbors, obtain DLCI neighbor addresses to add to the DLCI table. For information on DLCI addresses, see “How Frame Relay Works” later in this chapter.

Procedure

To configure transparent bridging over Frame Relay, follow these steps:

- 1 If your Frame Relay network does not support the LMI Protocol or if you want to bridge to specific DLCI neighbors, you must add DLCI neighbors to the static DLCI neighbor table using:

```
ADD !<port> -BRidge DlcINeighbor = <dlci>
```

Even if the LMI Protocol is enabled, you can add DLCI neighbors to the static DLCI neighbor table to bridge to specific DLCI neighbors. Static DLCI neighbors take precedence over neighbors learned dynamically with the LMI Protocol.

If LMI protocol is running consortium LMI, the valid range for subscriber numbers is 16 to 1022. For other LMI protocols, the range is 16 to 991.

- 2 Verify that transparent bridging has been enabled for the appropriate wide area port or virtual port by entering:

```
SHow -BRidge TRANSPARENTBRIDGE
```

By default, transparent bridging is enabled on all ports. If transparent bridging has been disabled for the wide area port, you can enable it using:

```
SETDefault !<port> -BRidge TRANSPARENTBRIDGE = TRANSPARENTBRIDGE
```

- 3 Verify that bridging is enabled by entering:

```
SHow -BRIDGE CONFIguration
```

If bridging has been disabled, enable it for the system by entering:

```
SETDefault -BRIDGE CONTROL = Bridge
```

Configuring Source Route Bridging

This section provides information for configuring source route bridging over Frame Relay.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in *Using Enterprise OS Software*.
- Set up the Frame Relay service as described in "Setting Up the Frame Relay Service" earlier in this chapter.
- Assign a unique ring number for each remote network.
- Assign a bridge number for the bridge.

Procedure

To configure source route bridging over Frame Relay, follow these steps:

- 1 Assign each wide area port of each bridge/router that is attached to the Frame Relay network the ring number (hexadecimal) of the network it accesses.

Use:

```
SETDefault !<port> -SR RingNumber = <number> (1-4095) | 0x<number> (1-FFF)
```

You can enter the ring number in decimal or hexadecimal format. Precede the hexadecimal number with 0x.

- 2 Verify that source route bridging is enabled on the wide area port using:

```
SHow !<port> -SR SrcRouBridge
```

If source route bridging is disabled, enable it for your wide area port using:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```

- 3 If you want to run source route and transparent bridging on a NETBuilder II bridge/router, skip this step and go on to step 4. If you want to run source route bridging only on a NETBuilder II bridge/router, disable transparent bridging on the wide area port using:

```
SETDefault !<port> -BRIDGE TransparentBRIDGE = NoTransparentBRIDGE
```

This step does not apply to model 32x and 52x SuperStack II bridge/routers. Transparent bridging is not supported on these models.

- 4 Verify that bridging is enabled by entering:

```
SHow -BRIDGE CONFIGURATION
```

If bridging has been disabled, enable it for the system by entering:

```
SETDefault -BRIDGE CONTROL = Bridge
```

Setting Up Basic Routing over Frame Relay

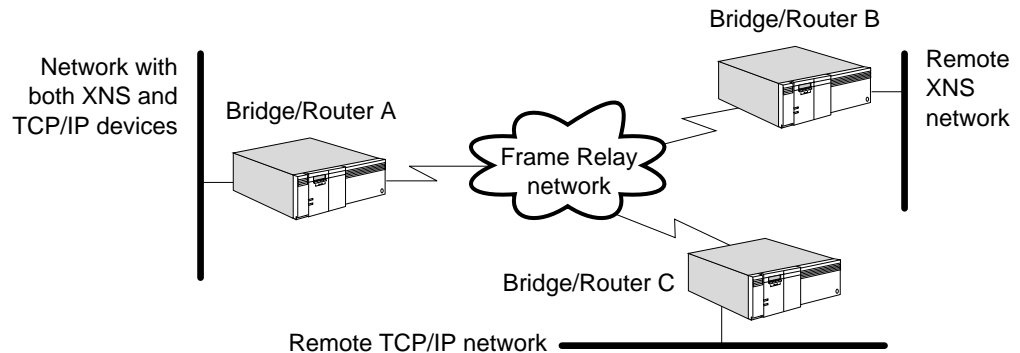
This section describes how to configure your router to transmit and receive data over a Frame Relay interface. Procedures for the following routing protocols are provided:

- AppleTalk
- APPN
- IPX
- OSI

- DECnet
- IP
- VINES
- XNS

A router can be configured to simultaneously route multiple protocols over Frame Relay to one or more remote network connections. For example, in Figure 339, the local network supports both XNS and TCP/IP traffic and routes information through a single Frame Relay connection to both types of remote networks.

Figure 339 Routing Multiple Protocols over Frame Relay Network



Configuring AppleTalk

This section provides information for configuring AppleTalk routing for communication over a Frame Relay network.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in *Using Enterprise OS Software*.
- Set up the Frame Relay service as described in "Setting Up the Frame Relay Service" earlier in this chapter.
- Determine whether to operate the Frame Relay network as either a non-AppleTalk or AppleTalk network. In both cases, Routing Table Maintenance Protocol (RTMP) packet broadcasts are sent as directed broadcasts every 10 seconds (this is the default) to each neighboring router configured on a port.

For a non-AppleTalk network configuration, obtain the Frame Relay DLCI addresses representing the virtual circuits to the routers at the remote networks so that you can configure static mapping.

For an AppleTalk network configuration, obtain the tentative network number and tentative node ID for each of the remote router ports connected to the Frame Relay network. Also obtain the Frame Relay DLCI addresses representing the virtual circuits to the routers at the remote networks so that you can configure static mapping.

For Frame Relay ports, split horizon decisions are made at the next router link level instead of at the port level when more than one neighbor link is associated with a port. Next-hop split horizon allows for support of partially meshed and nonmeshed topologies by allowing a router to use a Frame Relay port as a virtual hub, sending route information to each router out of the port learned from all other routers out of the same port. If the decisions were made at the port level, as is the case for AppleTalk on LANs and Switched Multimegabit Data Service

(SMDS), no routing information learned from any router out of the port would be sent to any router out of the same port.

Non-AppleTalk Configuration

To configure AppleTalk routing over a Frame Relay network configured as a non-AppleTalk network, see Figure 340 and follow these steps:

- 1 Configure all the ports on bridge/routers connected to the Frame Relay network to be connected to a non-AppleTalk network.

On bridge/routers A, B, and C, enter:

```
SETDefault !3 -AppleTalk CONTROL = NonAppleTalk
```

- 2 On each bridge/router, assign the Frame Relay DLCI of the other bridge/routers' ports and virtual ports connected to the network.

For example, on bridge/router A, enter:

```
ADD -AppleTalk ADDRESS !3 @33
ADD -AppleTalk ADDRESS !3 @44
```

Enter similar address information on bridge/routers B and C.

You can dynamically add and delete neighbors using the ADDRESS parameter while a port is enabled and AppleTalk is routing.

- 3 Enable routing on each AppleTalk bridge/router port attached to the Frame Relay network by entering:

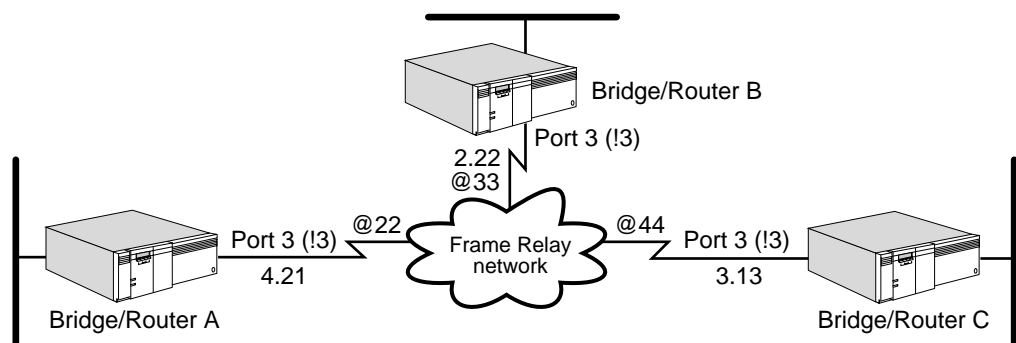
```
SETDefault !3 -AppleTalk CONTROL = ROUTE
```

AppleTalk Configuration

To configure AppleTalk routing over a Frame Relay network as an AppleTalk configuration, see Figure 340 and follow these steps.

The example in the following procedure assumes that the network range for the Frame Relay cloud shared by the configured routers is 2 to 4 and that at least one router is configured to send seed information to any other nonseed routers.

Figure 340 Configuring AppleTalk over Frame Relay



- 1 Specify the tentative network number and the tentative node ID for the specified port that the AppleTalk router uses during dynamic node address acquisition at port enable time.

Use:

```
SETDefault !<port> -AppleTalk StartupNET = <number>(0-65279)
SETDefault !<port> -AppleTalk StartupNODE = <number>(0-253)
```

Using these parameters allows the local router always to assign the same AppleTalk node address to the local port, assuming that the address is within the network range assigned to the Frame Relay cloud. These static configurations are saved nonvolatile storage and only need to be changed when the topology changes.

a For example, before routing is enabled on bridge/router A, enter:

```
SETDefault !3 -AppleTalk StartupNET = 4
SETDefault !3 -AppleTalk StartupNODE = 21
```

b Enter values for the StartupNET and StartupNODE parameters for bridge/routers B and C.

2 Configure static mapping of neighbor DLCIs to their AppleTalk node addresses on the ports and virtual ports of each bridge/router.

For example, on bridge/router A (AppleTalk address 4.21), enter the following DLCI addresses of the other routers connected to the Frame Relay network:

```
ADD -AppleTalk ADDRESS 2.22 @33
ADD -AppleTalk ADDRESS 3.13 @44
```

Configure static mapping of media addresses on bridge/routers B and C.

The valid range for Frame Relay DLCIs is 16 to 991 for user permanent virtual circuits.

You can dynamically add and delete neighbors using the ADDRESS parameter.

3 Enable routing on each AppleTalk bridge/router port attached to the Frame Relay network by entering:

```
SETDefault !3 -AppleTalk CONTROL = ROUTE
```

Configuring APPN

This section provides information for configuring the Advanced Peer-to-Peer Networking (APPN) network node for communication over a Frame Relay network.

You can configure APPN over Frame Relay over logical ports and over virtual ports. If you plan to send APPN traffic only over the port, use logical ports. Use virtual ports only if you plan to send APPN traffic and other protocol traffic over the same path to the same DLCIs. If you plan to use virtual ports, see "Configuring APPN with Virtual Ports" later in this chapter.

Prerequisites

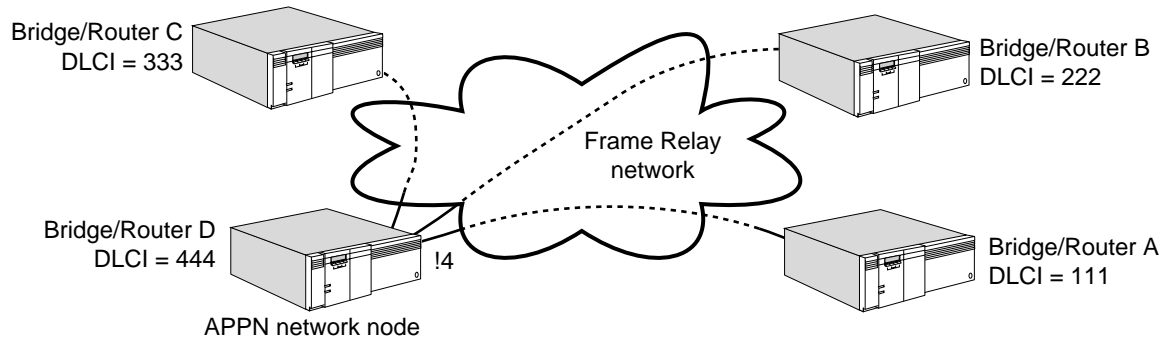
Before beginning this procedure, complete the following tasks:

- Set up the Frame Relay service as described in "Setting Up the Frame Relay Service" earlier in this chapter.
- Obtain the Frame Relay DLCI addresses of the remote networks to set up mapping information.

Procedure

To configure APPN to operate over a Frame Relay network, see Figure 341 and follow these steps:

Figure 341 Configuring APPN over Frame Relay



- 1 On bridge/router D, if the ports you plan to use to send APPN over Frame Relay are active APPN ports, deactivate each one using the following syntax and specify deactivate:

```
SET !<port> -APPN PortControl = (<Activate [NoLinkStations] | Deactivate [Orderly | Immediate]>)
```

- 2 Define each APPN port that is connected to the Frame Relay network using:

```
SETDefault !<port> -APPN PortDef = <DLC type>(LLC2|FR|DLSW|UNdef)
<max_btu_size>(99-8192) [ActLimit=<limit>(1-512)] [TGprof=<name>]
```

Make sure you specify Frame Relay as the data link control (DLC) type. For example, to configure port 2 for Frame Relay with a maximum basic transmission unit (BTU) size of 2,057, enter:

```
SETDefault !2 -APPN PortDef = FR 2057
```

If you are using logical ports, proceed to the next step. If you are using virtual ports, see “Configuring APPN with Virtual Ports” next before proceeding.

- 3 Configure the adjacent link stations for the Frame Relay logical port using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn) <max_btu_size>(99-8912)
<[Cmac|Ncmac] dest media addr> [Sap=<num>] [CPName=[netid.]cpname]
[Nodeid=<ID>] [LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
[CPSess=(Yes|No)]
```

For the destination media address, specify the destination DLCI number. For example, if you are configuring port 4 on node D in the figure to set up a link to node A, then enter a DLCI of 111 for node A.

To configure nodes A, B, and C as adjacent link stations to node D, assuming a maximum BTU size of 2057 and a service access point (SAP) value of 08, enter:

```
ADD !4 -APPN AdjLinkSta NN 2057 111 Sap=08
ADD !4 -APPN AdjLinkSta NN 2057 222 Sap=08
ADD !4 -APPN AdjLinkSta NN 2057 333 Sap=08
```

Repeat this step for each APPN port on bridge/router A that will communicate with DLCIs on the Frame Relay network.



Because of memory storage utilization issues, do not set the maximum BTU size higher than 2057.

- 4 If you want to change the default link characteristics, configure any desired link characteristics using:

```
SETDefault -APPN AdjLinkStaChar = <LinkStation name> [EffectCap=<string>]
[ConnectCost=<0-255>] [ByteCost=<0-255>] [Security=<string>]
[PropDelay=<string>] [Usd1=<0-255>] [Usd2=<0-255>] [Usd3=<0-255>]
```

For more information on the AdjLinkStaChar parameter, see the APPN Service Parameters chapter in *Reference for Enterprise OS Software*.

- 5 To reduce the number of Logical Link Control, type 2 (LLC2) retries the system performs and the amount of time the LLC2 timer reply waits for a response to a test frame, change the values of the -LLC2 RetryCount and TimerInact parameters.

These steps are necessary because when you set the port owner as Frame Relay (using the -PORT OWner parameter), different default values are assigned to the -LLC2 RetryCount and TimerInact parameters. It will take seven minutes to discover a link outage. If you try to deactivate the local APPN network node when this happens, the network node will not be able to deactivate until the reply is received, delaying the deactivation for up to seven minutes. If the local bridge/router is trying to contact a remote bridge/router that is not available, it will take seven minutes for the local bridge/router to discover this. To prevent this long delay, reset the values for these two parameters by entering:

```
SETDefault -LLC2 RetryCount = 3
SETDefault -LLC2 TimerInact = 30000
```

By changing these two values, you will reduce the time required for this process to 90 seconds.

- 6 Activate the APPN ports using:

```
SET !<port> -APPN PortControl = (<Activate [NoLinkStations])
```

- 7 Repeat steps 1 through 6 on nodes B, C, and D.

To ensure connectivity between two partner network nodes, the adjacent link station configuration should be performed on both sides.

You can fully mesh a configuration similar to the one shown in Figure 341 without using virtual ports.

Configuring APPN with Virtual Ports

You normally do not need to use virtual ports to configure APPN to operate over Frame Relay. The purpose of virtual ports is to enable multiple ports to be active on the same physical path. Because APPN allows multiple links to be active on a path at the same time, it provides the same type of capability that virtual ports provide. However, if you want to send APPN data and other protocols over the same physical path to a Frame Relay network, you may need to use virtual ports.



CAUTION: *Configure virtual ports before configuring the APPN network node.*

Prerequisite Configure the virtual port using the procedures described in the Configuring Advanced Ports and Paths chapter.

Procedure Follow the procedure described in the previous section. However, when you configure adjacent link stations in step 3, use virtual ports. Configure the AdjLinkSta parameter as you normally would, but specify a virtual port instead of the logical port.

For example, to add a link from bridge/router D to bridge/router C using virtual port 4 with a maximum BTU size of 2057 and a SAP/TCP value of 08, enter:

```
ADD !V4 -APPN AdjLinkSta NN 2057 333 Sap=08
```

After you configure the adjacent link stations, follow the remainder of the previous procedure.



You can configure virtual ports for adjacent link stations only if the DLC type for the PortDef parameter is set to FR. If the DLC type is not set to FR, the virtual ports will not be valid.

If you configure virtual ports, when you enter the command to display link stations, the virtual ports will display as logical ports. The !V designation will not be shown in the display.

Deleting APPN Virtual Ports

After you have configured virtual ports for APPN over Frame Relay, you must be careful when deleting them. If you delete virtual ports without first deleting the adjacent link stations associated with the virtual port, you will not be able to access the link station, and you will lose all sessions over that link station.

To delete virtual ports used for APPN, follow these steps:

- 1 Delete the destination adjacent link station the virtual port was using, specifying the link name:

```
DELEte !<port> -APPN AdjLinkSta <LinkName>
```

For example, to delete the adjacent link station with a link name of 00001 on virtual port 4, enter:

```
DELEte !V4 -APPN AdjLinkSta 00001
```

- 2 Deactivate the physical port and specify deactivate, making sure to also specify the logical port that was mapped to the virtual port:

```
SET !<port> -APPN PortControl = (<Activate [NoLinkStations] | Deactivate [Orderly | Immediate]>)
```

This command deactivates all active sessions being used by the logical port.

- 3 Delete the virtual port using:

```
DELEte !<port> -PORT VirtualPort {<path> {<FR_DLCI>}}
```

For more information, see "VirtualPort" in the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

- 4 Repeat this procedure for each virtual port being deleted.

Configuring DECnet This section provides information for configuring DECnet routing for communication over a Frame Relay network.

Prerequisites

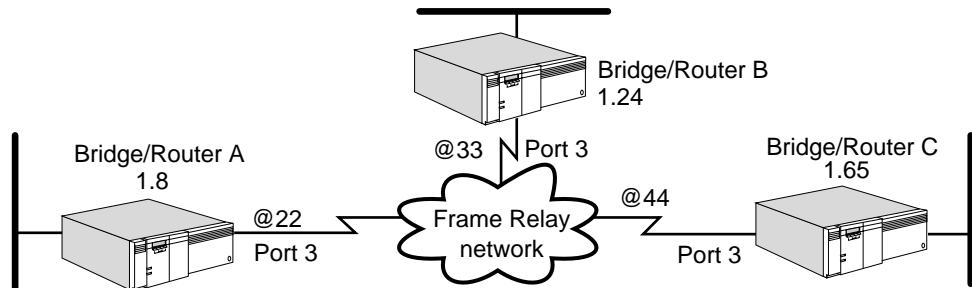
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in *Using Enterprise OS Software*.
- Set up the Frame Relay service as described in "Setting Up the Frame Relay Service" earlier in this chapter.
- Obtain the DECnet addresses and the Frame Relay DLCI addresses of the remote networks to set up mapping information.

Procedure

To configure DECnet routing over a Frame Relay network, see Figure 342 and follow these steps:

Figure 342 Configuring DECnet over Frame Relay



- 1 Specify DECnet-to-FR address mapping information on each port or virtual port that is attached to the Frame Relay network.

For example, on port 3 of bridge/router A, enter:

```
ADD !3 -DECnet Neighbor 1.24 @33
ADD !3 -DECnet Neighbor 1.65 @44
```

On bridge/routers B and C, specify the DECnet-to-Frame Relay address mapping information.

- 2 Enable DECnet routing on each port of each bridge/router that is attached to the Frame Relay network.

For example, to enable routing on port 3 of bridge/router A, enter:

```
SETDefault !3 -DECnet CONTROL = ROute
```

Enable routing on bridge/routers B and C.

This completes the procedure for configuring DECnet routing over a Frame Relay network.

Configuring IP

This section provides information for configuring IP routing for static and dynamic address resolution over a Frame Relay network. If your network is small and relatively stable, 3Com recommends that you configure the `-ARP CONTROL` parameter with the `NoInArp` value. This static address resolution reduces network overhead during initialization.

If your network is large and needs to be reconfigured frequently, 3Com suggests that you configure the `-ARP CONTROL` parameter with the `InARP` value. This dynamic configuration can save you some network administration work. InARP entries in the IP address table are learned when:

- IP addresses are configured.
- A new DLCI is available.

InARP entries in the IP address table are deleted when:

- The IP address table is flushed. After this occurs, InARP immediately sends out InARP requests and discovers new entries.
- An existing DLCI becomes unavailable.

To minimize the network overhead, once an IP address associated with a specific DLCI is discovered, it is treated as a static entry and is not aged out.

Prerequisites

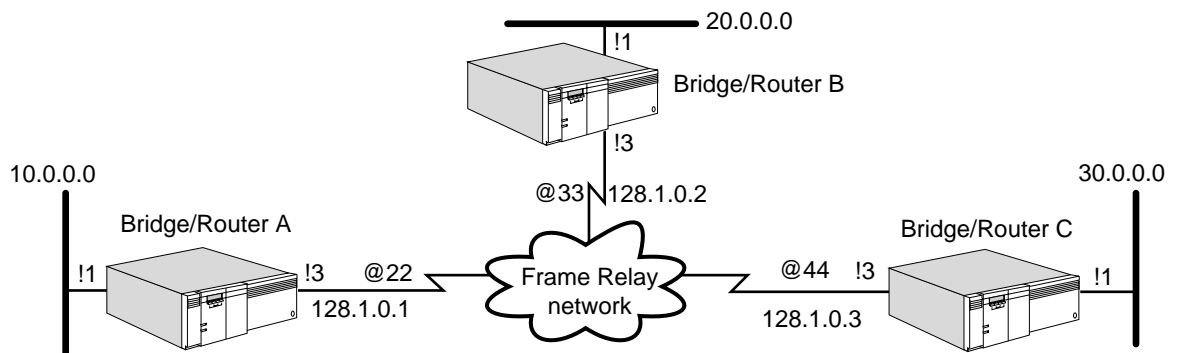
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in the IPv6 Service Parameters chapter.
- Set up the Frame Relay service as described in “Setting Up the Frame Relay Service” earlier in this chapter.
- Determine the IP addresses for each wide area port of your bridge/router that is attached to the Frame Relay network. If using virtual ports, each virtual port must have a unique IP address and be on a separate IP subnet.
- For static configurations only, obtain the Frame Relay DLCI addresses to create IP-to-FR mappings for static routes.
- Obtain the IP addresses of neighbors that should send or receive RIP update packets.

Procedure

To enable IP to operate over a Frame Relay network with a static or dynamic configuration, see Figure 343 and follow these steps:

Figure 343 Configuring IP over Frame Relay



If you are configuring your network for static address resolution over a Frame Relay network, make sure you have set the `-ARP CONTROL` parameter to the `NoInArp` value by using the `SETDefault !<port> -ARP CONTROL = NoInArp` syntax.

- 1 Assign an IP address to each port or virtual port on each NETBuilder bridge/router that is directly attached to the Frame Relay network.

For example, to assign the address 128.1.0.1 to port 3 on bridge/router A, enter:

```
SETDefault !3 -IP NETaddr = 128.1.0.1
```

- 2 For static address resolution over a Frame Relay network, specify the IP-to-Frame Relay DLCI address mapping information. Specify the IP-to-Frame Relay DLCI address mapping information for each bridge/router connected to a Frame Relay network to which the system wants to communicate.



Do not perform this step if you are configuring the network for dynamic address resolution over a Frame Relay network. Proceed to step 3.

Using Figure 343 as an example, the following sequence of commands specify IP-to-Frame Relay DLCI mapping information for the routers directly attached to the Frame Relay network. The valid range for Frame Relay DLCIs is 16 through 991

for user permanent virtual circuits. (In the examples that follow, DLCI can be used in place of @.)



You must specify this information for DLCIs associated with ports as well as virtual ports.

For example, on bridge/router A (IP address 128.1.0.1) enter:

```
ADD -IP ADDRESS 128.1.0.2 @22
ADD -IP ADDRESS 128.1.0.3 @22
```

Enter similar commands on bridge/router B (IP address 128.1.0.2) and bridge/router C (IP address 128.1.0.3), specifying the IP address and DLCI mapping information.

- 3 For dynamic address resolution over a Frame Relay network, enable the ARP Service to automatically discover IP addresses for the DLCIs.

For example, to enable InArp on port 3, enter:

```
SETDefault !3 -ARP CONTROL = InArp
```

- 4 Add each bridge/router on a Frame Relay network to which the system wants to communicate with as a neighbor.



You can skip this step if the DynamicNbr option of the -RIP and -OSPF CONTROL parameters are enabled. If DynamicNbr is disabled, you must specify this information for ports as well as virtual ports.

Complete the following steps:

- a Specify a list of neighbor addresses to which RIP will send update packets. For example, to transmit RIP packets from bridge/router B, which is running RIP, to bridge/router C, enter:

```
ADD !3 -RIP AdvToNeighbor 128.1.0.3
```

- b Add IP addresses of neighbors on each bridge/router port that is participating in RIP.
- c Do not change the default for all neighbors.
- d Specify a list of neighbor addresses to which OSPF will send update packets.

For example, to transmit OSPF packets from bridge/router B, which is running OSPF, to bridge/router C, enter:

```
ADD !3 -OSPF Neighbor 128.1.0.3
```

- e Add IP addresses of neighbors on each bridge/router port that is participating in OSPF.
- 5 Enable the dynamic routing protocols for IP using RIP, OSPF, or Integrated IS-IS (ISIS) for each port and/or virtual port.
 - To learn routes dynamically on port 3 using RIP, determine if the Frame Relay network is fully meshed or nonmeshed. If fully meshed, enter:

```
SETDefault !3 -RIP CONTROL = (Talk, Listen, FullMesh)
```

If nonmeshed, enter:

```
SETDefault !3 -RIP CONTROL = (Talk, Listen, NonMesh)
```

Setting the CONTROL parameter to the TALK and Listen values enables the router to send and receive routing information with other routers using RIP.

- To enable routes dynamically on port 3 using OSPF, determine if the Frame Relay network is fully meshed or nonmeshed.

If nonmeshed, you must run NonMesh. Enter:

```
SETDefault !3 -OSPF CONTrol = (Enable, NonMesh)
```

If fully meshed, you must run FullMesh. Enter:

```
SETDefault !3 -OSPF CONTrol = (Enable, FullMesh)
```

All of the OSPF neighboring routers must be configured with the same mode: FullMesh or NonMesh. Both modes apply to ports as well as virtual ports.

Once OSPF operation is enabled, the router will exchange routing information with other routers using OSPF.

- To enable routes dynamically using IISIS, see the IPv6 Service Parameters chapter.
- 6 Verify that IP routing is enabled on each bridge/router that is attached to the Frame Relay network by entering:

```
SHow -IP CONFiguration
```

If IP routing has been disabled, enable it by entering:

```
SETDefault -IP CONTrol = ROute
```

This completes the procedure for configuring IP for communication over a Frame Relay network.

Configuring IPX

This section provides information for configuring IPX routing for communication over a Frame Relay network.

Prerequisites

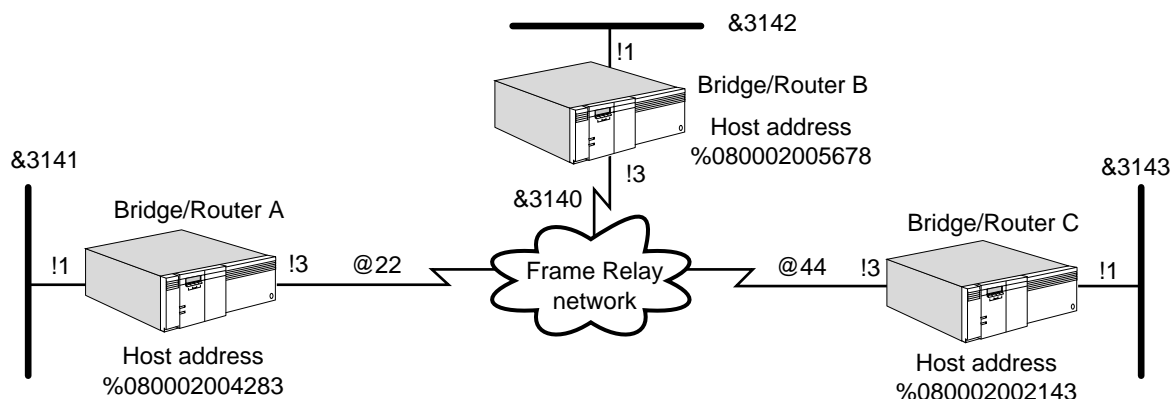
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in *Using Enterprise OS Software*.
- Set up the Frame Relay service as described in "Setting Up the Frame Relay Service" earlier in this chapter.
- Determine the IPX network numbers to be assigned to each wide area port attached to the Frame Relay network. If using virtual ports, each virtual port must have a unique network number.

Procedure

To configure IPX to operate over a Frame Relay network, see Figure 344 and follow these steps:

Figure 344 Configuring IPX over Frame Relay



- 1 Assign a network number to each port or virtual port on each NETBuilder bridge/router connected to the Frame Relay network.

For example, assign &3140 as the NETnumber to port 3 on bridge/routers A, B, and C by entering (on each router):

```
SETDefault !3 -IPX NETnumber = &3140
```

- 2 Obtain the Frame Relay DLCI addresses of the remote networks to set up mapping information.
- 3 You must set up mapping information between Frame Relay addresses and host addresses for each bridge/router directly connected to the Frame Relay network.

For example, on bridge/router A, enter:

```
ADD !3 -IPX ADDRESS @33 %080002005678
ADD !3 -IPX ADDRESS @44 %080002002143
```

The physical MAC addresses of the neighbors are optional. If you want to use the physical MAC addresses of the neighbors, you can obtain them by using the SHow -SYS ADDRESS command.

- 4 If you want the bridge/router to automatically send routing updates to all of the active data link connection identifiers (DLCIs), enable the DynamicNbr option in the NRIP, SAP, and NLSP CONTROL parameters. With DynamicNbr enabled, the router assumes every active DLCI points to another IPX router that is fully trusted.

If you want the bridge/router to exchange routing with only specific neighbors, disable the DynamicNbr option in the NRIP, SAP, and NLSP CONTROL parameters and configure each individual neighbor in the AdvToNeighbor parameter.

For example, on bridge/router A, to specify that bridge/router B receives route reachability information, enter:

```
ADD !3 -NRIP AdvToNeighbor &3140%080002005678
ADD !3 -SAP AdvToNeighbor &3140%080002005678
```

For NLSP, configure the Neighbor parameter for each neighboring router.

- 5 Specify the DLCI of neighbors that will be taking part in routing over Frame Relay using:

```
ADD !<port> -NLSP Neighbors @<DLCI>
```

For example on bridge/router A, enter the DLCIs of bridge/routers B and C:

```
ADD !3 -NLSP Neighbors @33
ADD !3 -NLSP Neighbors @44
```


- 6 Enable the use of policy parameters by entering:

```
SETDefault !3 -NRIP PolicyControl = AdvToNbr
SETDefault !3 -SAP PolicyControl = AdvToNbr
```

- 7 Verify that IPX routing is enabled on each bridge/router that is attached to the Frame Relay network by entering:

```
SHow -IPX CONFIguration
```

If routing has been disabled on bridge/router A, enable it by entering:

```
SETDefault !3 -IPX CONTRol = ROute
```

Enable routing on bridge/routers B and C.

- 8 If you are using NRIP and SAP as your routing protocols, verify that routing is enabled on each wide area port of each bridge/router that is attached to the Frame Relay network by entering:

```
SHow -NRIP CONTRol
```

Using the SHow -SAP CONTRol command, verify that Auto, Talk and Listen, or DynamicNbr (for non-broadcast multiaccess (NBMA) networks) are set.

- 9 If you are using NetWare Link Services Protocol (NLSP) as the routing protocol, complete the following steps:
- a If you are communicating to a non-3Com router over Frame Relay, enable the IpxWan option by entering:

```
SETDefault !3 -IPX CONTRol = IpxWan
```

- b Make sure the NLSP is enabled by entering:

```
SHow -NLSP CONTRol
```

- c Display the NLSP adjacencies by:

```
SHow -NLSP ADJAcencies
```

This completes the procedure for configuring IPX routing over a Frame Relay network.

Configuring OSI This section provides information for configuring OSI routing for communication over a Frame Relay network.

Prerequisites

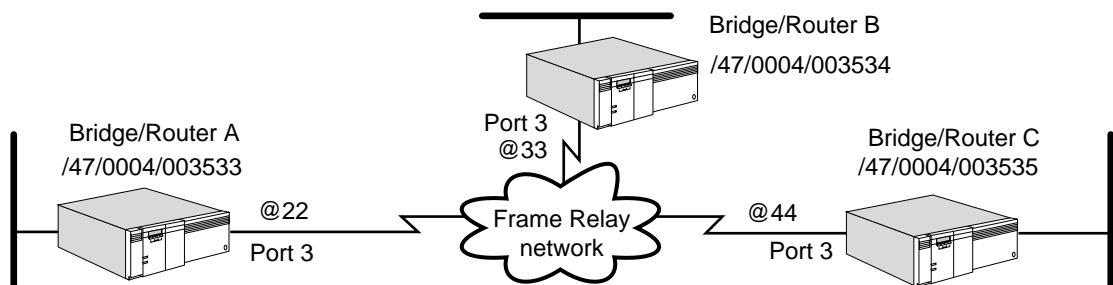
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in *Using Enterprise OS Software*.
- Set up the Frame Relay service as described in "Setting Up the Frame Relay Service" earlier in this chapter.
- Determine whether to use the PrefixRoute or the Neighbors parameter.
 - Use the PrefixRoute parameter if you view the remote site as another routing domain, for example, another company, with a different NSAP address. The PrefixRoute parameter allows you to specify interdomain reachability information without exchanging IS-IS packets.
 - Use the Neighbors parameter if the remote site is part of your routing domain. The neighbor information instructs the IS-IS Protocol to exchange packets and establish full connectivity.

Procedure

To configure OSI to operate over a Frame Relay network, see Figure 345 and follow these steps. If you are configuring the PrefixRoute parameter, begin with step 1. If you are configuring the Neighbors parameter, begin with step 2.

Figure 345 Configuring OSI over Frame Relay



- 1 Specify the OSI network service access point (NSAP) prefix and corresponding Frame Relay address for static interdomain routing across the Frame Relay network.

Use the `-ISIS PrefixRoute` parameter. The `-ISIS MODE` parameter must be set to L2 for the `PrefixRoute` parameter to take effect.

Set up static interdomain routing on bridge/router A by entering:

```
ADD !3 -ISIS PrefixRoute /47/0004/003534 @33
ADD !3 -ISIS PrefixRoute /47/0004/003535 @44
```

Specify OSI-to-FR address mapping information on bridge/routers B and C.

Proceed to step 3.

- 2 Specify neighbors on the Frame Relay network that support IS-IS for dynamic intradomain routing.

For example, from bridge/routers A and C enter:

```
ADD !3 -ISIS Neighbors @33
ADD !3 -ISIS Neighbors @44
```

Repeat this step for bridge/routers B and C.

- 3 Verify that ISIS routing is enabled on each bridge/router that is attached to the Frame Relay network by entering:

```
SHow -CLNP CONFIguration
```

If routing has been disabled, enable it on bridge/router A by entering:

```
SETDefault -CLNP CONTrol = Route
```

Enable routing on bridge/routers B and C.

This completes the procedure for configuring OSI routing over a Frame Relay network.

Configuring VINES

This section provides information for configuring VINES routing for communication over a Frame Relay network.

Prerequisites

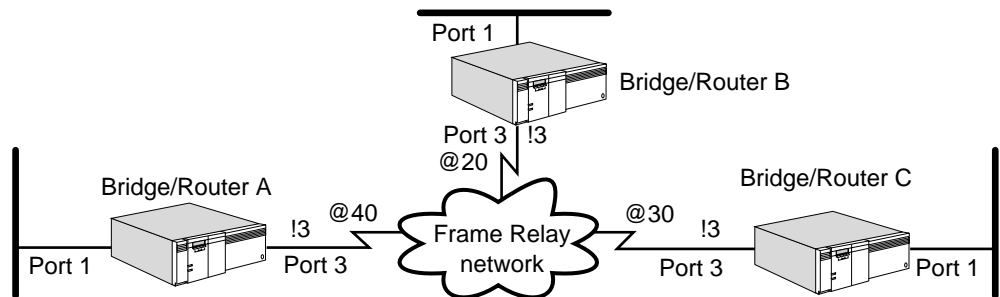
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in *Using Enterprise OS Software*.
- Set up the Frame Relay service as described in "Setting Up the Frame Relay Service" earlier in this chapter.
- Obtain the Frame Relay DLCI addresses of the remote networks.

Procedure

To enable the VINES Protocol to operate over a Frame Relay network, see Figure 346 and follow these steps:

Figure 346 Configuring VINES over Frame Relay



- 1 Specify Frame Relay DLCI addresses for ports or virtual ports.

For example, to specify the DLCI address for port 3 on bridge/router A, enter:

```
ADD !3 -VIP WideAreaNbr @20
ADD !3 -VIP WideAreaNbr @30
```

On bridge/routers B and C, specify the DLCI addresses for the ports.

- 2 Verify that VINES routing is enabled on each bridge/router that is attached to the Frame Relay network by entering:

```
SHOW -VIP CONFIGuration
```

If routing has been disabled, enable it on bridge/router A by entering:

```
SETDefault -VIP CONTROL = Route
```

Enable routing on bridge/routers B and C.

Configuring XNS

This section provides information for configuring XNS routing for communication over a Frame Relay network.

Prerequisites

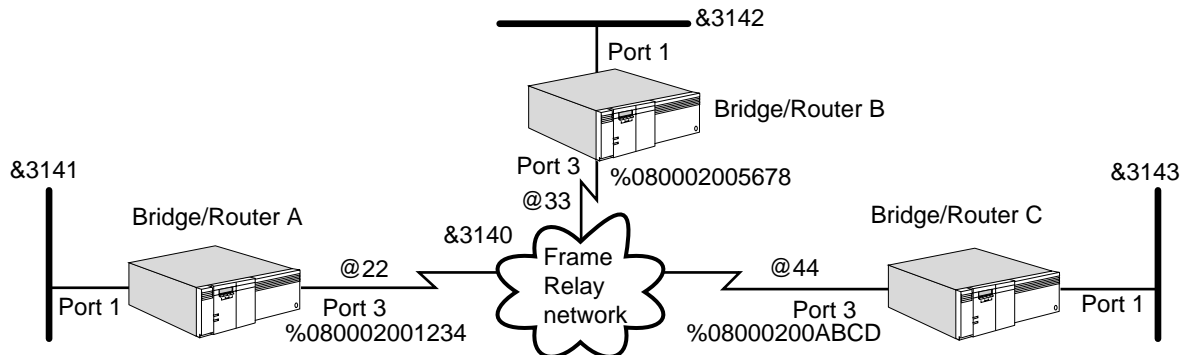
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in *Using Enterprise OS Software*.
- Set up the Frame Relay service as described in "Setting Up the Frame Relay Service" earlier in this chapter.
- Determine the XNS network number for each wide area port of your bridge/router that is attached to the Frame Relay network. If using virtual ports, each virtual port must have a unique network number.
- Obtain the media access control (MAC) addresses and Frame Relay DLCI addresses of the remote networks to set up mapping information.

Procedure

To enable the XNS Protocol to operate over a Frame Relay network, see Figure 347 and follow these steps:

Figure 347 Configuring XNS over Frame Relay



- 1 Assign a NETnumber to each port or virtual port on each bridge/router that is connected to the Frame Relay network.

For an example, assign &3140 as the NETnumber to port 3 on bridge/routers A, B, and C by entering (on each router):

```
SETDefault !3 -IDP NETnumber = &3140
```

- 2 If your Frame Relay network supports the LMI Protocol and you selected the appropriate version of the protocol as described in "Setting Up the Frame Relay Service" earlier in this chapter, skip this step and go on to step 3. If your Frame Relay network does not support the LMI Protocol and you disabled this protocol as described in "Setting Up the Frame Relay Service," set up mapping information between Frame Relay addresses and host addresses for each bridge/router directly connected to the Frame Relay network.

For example, on bridge/router A, enter:

```
ADD !3 -RIPXNS ADDRESS %080002005678 @33
```

Set up mapping information on bridge/routers B and C.

- 3 Verify that IDP routing is enabled on each bridge/router that is attached to the Frame Relay network by entering:

```
SHow -IDP CONFIGuration
```

If IDP routing has been disabled, enable it by entering:

```
SETDefault -IDP CONTrol = Route
```

Enable routing on bridge/routers B and C.

Configuring Disaster Recovery

This section discusses how to configure disaster recovery in a Frame Relay environment. The information in this section applies only to platforms that support the configuration of virtual ports.

Disaster recovery is a mechanism that allows you to maintain connectivity between your central and remote sites in the event of failure of a physical line or the Frame Relay network. This feature provides a way to recover from the loss of a primary permanent virtual circuit (PVC) in a Frame Relay network by triggering a backup PVC. If the primary PVC becomes unavailable, as determined by the LMI Protocol,

the traffic destined for the primary PVC is forwarded over the backup PVC, maintaining connectivity between nodes. Upon recovery of the primary PVC, the backup PVC is deactivated, and traffic is again forwarded over the primary PVC. The backup PVC can be configured on a separate link to provide redundancy. The backup link can be either a leased or dial-up link.



For conceptual information on how disaster recovery works, see “How Disaster Recovery Works” later in this chapter.

Prerequisites Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your wide area interfaces.
- Configure virtual ports.
- Configure the primary link according to “Setting Up the Frame Relay Service” earlier in this chapter.
- Acquire services from a Frame Relay service provider. For more information, see the *WAN Cabling and Connectivity Guide*. You can find this guide on the 3Com Corporation World Wide Web site by entering:

<http://www.3Com.com/>

Procedure The following procedures describe how to configure a primary line, a backup PVC, a backup PVC on a separate link, and a backup PVC on a separate dial-up link in the event of the failure of a primary line. Figure 348 is a configuration example.

Figure 348 Fully Redundant Network Among All Sites

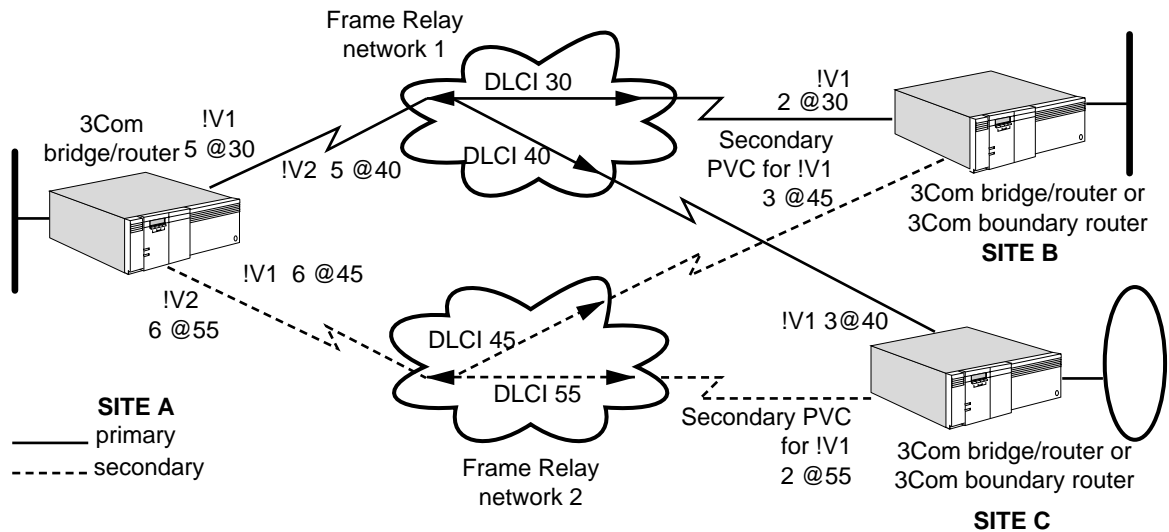


Figure 348 shows a configuration that is fully redundant between site A, the central site, and remote sites B and C. At site A, the primary PVCs are on link 5, and the backup PVCs are on link 6. Both of these lines are leased lines. At site B, the primary PVC is on link 2, and the backup link is on link 3. Link 3 at site B is a dial-up line. At site C, the primary PVC is on link 3, and the backup PVC is on a dial-up link 2.

Configuring a Primary PVC

To configure a primary PVC, you must first set up a virtual port with that PVC. Before setting up virtual ports, make sure the owner of the wide area port associated with the path through which the virtual ports will be defined is set to Frame Relay. Create a virtual port for each remote network that is attached to a Frame Relay cloud. You also must configure the primary link according to “Setting Up the Frame Relay Service” earlier in this chapter.

To configure a primary PVC, see Figure 348 and follow these steps:

- 1 Configure the primary PVC between site A and site B.

- a Set up a virtual port for site A by entering:

```
ADD !V1 -PORT VirtualPort 5@30
```

This command designates 5@30 as a primary PVC.

- b Set up a virtual port for site B by entering:

```
ADD !V1 -PORT VirtualPort 2@30
```

This command designates 2@30 as a primary PVC.

When creating virtual ports, you must designate the same PVC on both ends of the connection as primary.

- 2 Configure the primary PVC between site A and site C.

- a Set up a virtual port for site A by entering:

```
ADD !V2 -PORT VirtualPort 5@40
```

This command designates 5@40 as a primary PVC.

- b Set up a virtual port for site B by entering:

```
ADD !V1 -PORT VirtualPort 3@40
```

This command designates 3@40 as a primary PVC.

When creating virtual ports, you must designate the same PVC on both ends of the connection as primary.

Configuring a Backup PVC

To configure a backup PVC for disaster recovery, see Figure 348 and follow these steps:

- 1 Configure the backup PVC from site A to site B by entering:

```
ADD !V1 -FR BackupPVC 6@45
```

- 2 Configure the backup PVC from site B to site A by entering:

```
ADD !V1 -FR BackupPVC 3@45
```

You must designate the same PVC on both ends of the connection between sites A and B as backup.

- 3 Configure the backup PVC from site A to site C by entering:

```
ADD !V2 -FR BackupPVC 6@55
```

- 4 Configure the backup PVC from site C to site A by entering:

```
ADD !V1 -FR BackupPVC 2@55
```

You must designate the same PVC on both ends of the connection between sites A and C as backup.

By default, the port is brought down when the primary PVC fails, even when the backup PVC is available.

Configuring a Backup Link

You can add a backup PVC to a previously configured virtual port to provide redundancy. If the backup PVC is on a separate path, this path must be attached to a separate port.

To configure a backup link see Figure 348 and follow these steps:

- 1 Enable the Frame Relay Service by setting the owner of the serial interface to Frame Relay by entering:

```
SETDefault !6 -PORT OWNeR = FrameRelay
```

- 2 If your Frame Relay network supports the LMI Protocol, make sure that the appropriate LMI Protocol is enabled. For more information about enabling the LMI Protocol, see "Setting Up the Frame Relay Service" earlier in this chapter of this chapter.

To manually enable the LMI Protocol, enter:

```
SETDefault !6 -FR CONTroL = LMI
```

How Frame Relay Works

This section provides the following basic information about Frame Relay networks:

- A description of permanent virtual circuits (PVCs) and switched virtual circuits (SVCs)
- Definitions of fully meshed, partially meshed, and nonmeshed Frame Relay topologies and solutions to work around the connectivity problems that partially meshed and nonmeshed topologies present
- Frame Relay addresses
- LMI Protocol
- Disaster recovery over Frame Relay
- Partially and fully redundant Frame Relay networks
- A description of the Frame Relay Congestion Control feature
- Frame Relay auto startup

The wide area bridge/router supports both bridging and routing of multiple protocols over Frame Relay. ATM, X.25, and SMDS allow only one path to be assigned to a port, which means only one of these wide area protocols can run over a path or serial line. The running of only one wide area protocol over a path or serial line is mostly true of Frame Relay; however, it is possible to configure dual permanent virtual circuits (PVCs) bound to a single logical port on a boundary router. Figure 349 shows the adding of a backup PVC on path 3 for logical port 2.

To add a backup PVC on path 3 for logical port 2, follow these steps:

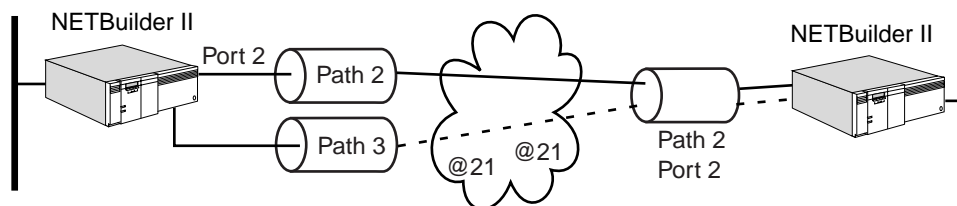
- 1 Configure port 2 of the NETBuilder II bridge/router number 1 for Frame Relay over a backup PVC 3 at DLCI 21, by entering:

```
ADD !2 -FR BackupPVC 3@21
```

- 2 Configure port 2 of the NETBuilder II bridge/router number 2 for Frame Relay over a backup PVC 2 at DLCI 21, by entering:

```
ADD !2 -FR BackupPVC 2@21
```

Figure 349 Configuring Dual PVCs Over a Single Logical Port



Frame Relay allows your bridge/router to transmit and receive data from any other device on the Frame Relay network over a PVC and over an SVC, which provide virtual connections to all other nodes on the network.

PVC and SVC Connections

PVCs are permanent circuits between the remote user and the local bridge/router. SVCs are dynamic circuits set up between the remote user and the local bridge/router, which are disconnected after data activity has stopped. You can also configure an SVC as a static connection that works like a PVC by setting the -FR SvcdIdleTimer setting to zero or none. See the FR Service Parameters chapter in *Reference for Enterprise OS Software* for details.

PVCs use the data link circuit identifier (DLCI) numbers assigned by the Frame Relay service provider to establish the connection between the remote user and the local NETBuilder bridge/router logical or virtual port. The DLCI numbers assigned by the service provider come from a range of DLCI numbers available.

The range of DLCI numbers can range from 0 to 1023. The service provider assigns a subset from this range and reserves the rest. The range assigned depends on which protocol the bridge/router port was configured for:

- For ports configured for ANSI or NTLMI, the service provider assigns DLCI numbers ranging from 16 to 991. Numbers 0 to 15 and 992 to 1023 are reserved and not assigned.
- For ports configured for other LMI protocols, the service provider assigns numbers ranging from 16 to 998. Numbers 0 to 15 and 999 to 1023 are reserved and not assigned.

When configuring SVCs, a virtual circuit identifier <vcid> number is assigned for the SVC that is mapped to a number from the numbers in the DLCI range that were not assigned for PVCs by the service provider. For example, if the service provider assigned DLCI numbers 16 through 50, 85, 87, and 89 for PVCs from the range of 16 to 991, you can assign vcid numbers for SVCs from the other numbers still available within the range of 16 to 991. Although an SVC is assigned a virtual circuit identifier number that maps to a specific DLCI number, because SVCs are dynamic, any virtual circuit available (other than those assigned for PVCs) when a connection is needed is used by the SVC to establish the connection.

Establishing a PVC Connection

For PVCs, the connection between the remote user and the local bridge/router port through the Frame Relay network is established when you enter the DLCI number assigned by the service provider. You enter the DLCI number when you configure the individual protocol that will transmit through the Frame Relay network.

For example, to establish a PVC connection using AppleTalk protocol over Frame Relay using DLCI 205, and AppleTalk node address 50.210, enter:

```
ADD -AppleTalk ADDRESS 50.210 @ 205
```

Establishing an SVC Connection

Unlike when you establish a PVC connection, you must follow several steps to establish an SVC connection between a remote user and the local bridge/router port through the Frame Relay network.

Included in these steps is entering a virtual circuit identifier (vcid) number that identifies the SVC. The vcid number must match a number from the range of DLCI numbers available and not assigned by the service provider for PVCs.

To establish an SVC connection, follow these steps:

- 1 When configuring the individual protocol that will transmit through Frame Relay, enter the vcid number that identifies the SVC used for the connection.

For example, to establish an SVC for a network connection that:

- uses AppleTalk protocol over Frame Relay
- has an AppleTalk node address of 70.220
- uses vcid number 300 to identify the SVC (from the list of DLCI numbers not assigned for a PVC)

enter:

```
ADD -AppleTalk ADDRESS 70.220 @ 300
```

- 2 Identify the local customer premises equipment (CPE) link address (telephone number) assigned by your Frame Relay network service provider for your NETBuilder bridge/router, which identifies the local bridge/router port to the network, using:

```
SETDefault !<port> -FR LinkAddress = [<E.164 address> (1-15 digits) | <X.121 address> (1-15 digits)]
```

The -FR LinkAddress parameter identifies the local bridge/router port for all SVCs configured for that port.

- 3 Enter the destination address (the telephone number) of the remote user associated with the bridge/router virtual circuit ID using:

```
ADD !<port> -FR SvcDestAddress = <vcid> [<E.164 address> (1-15 digits) | <X.121 address> (1-15 digits)]
```

where <vcid> specifies a virtual circuit identifier number that is mapped to a DLCI number from the range of numbers available that have not been assigned by your service provider for a PVC connection.

- 4 Identify the local SVC telephone number associated with a specified virtual circuit ID (<vcid>) to the remote user (optional) using:

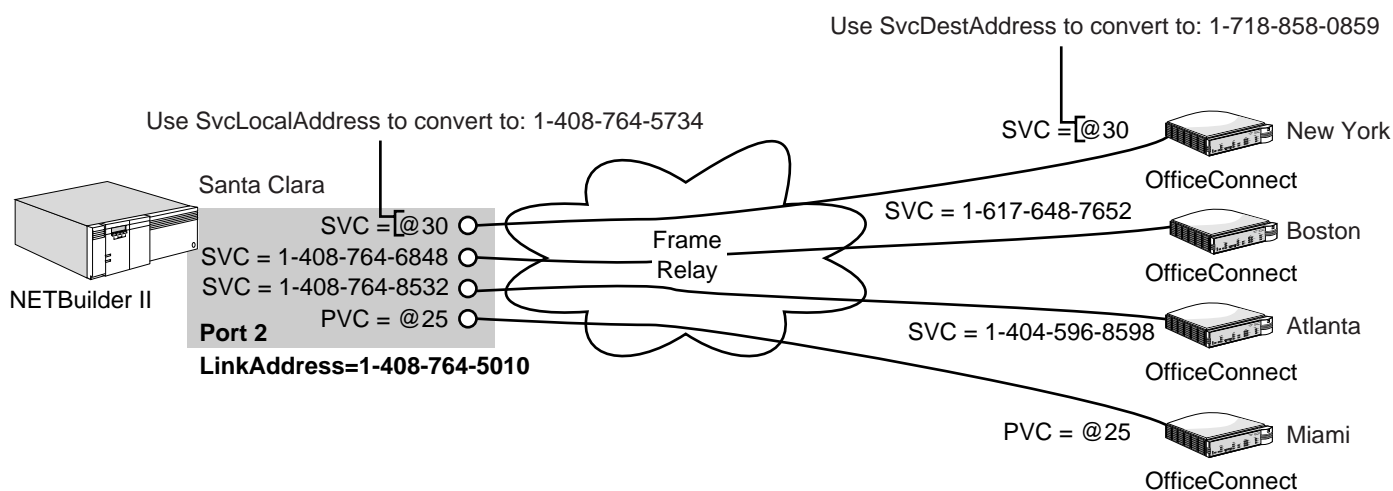
```
SETDefault !<port> -FR SvcLocalAddress = <vcid> [<E.164 address> (1-15 digits) | <X.121 address> (1-15 digits)]
```

where <vcid> specifies a virtual circuit identifier number that is mapped to a DLCI number from the range of numbers available that have not been assigned by your service provider for a PVC connection. The CPE link address is used to identify the local virtual circuit if the address of the local SVC is not set.

Figure 350 shows NETBuilder bridge/router port 2 (in Santa Clara), over Frame Relay, connecting a remote user:

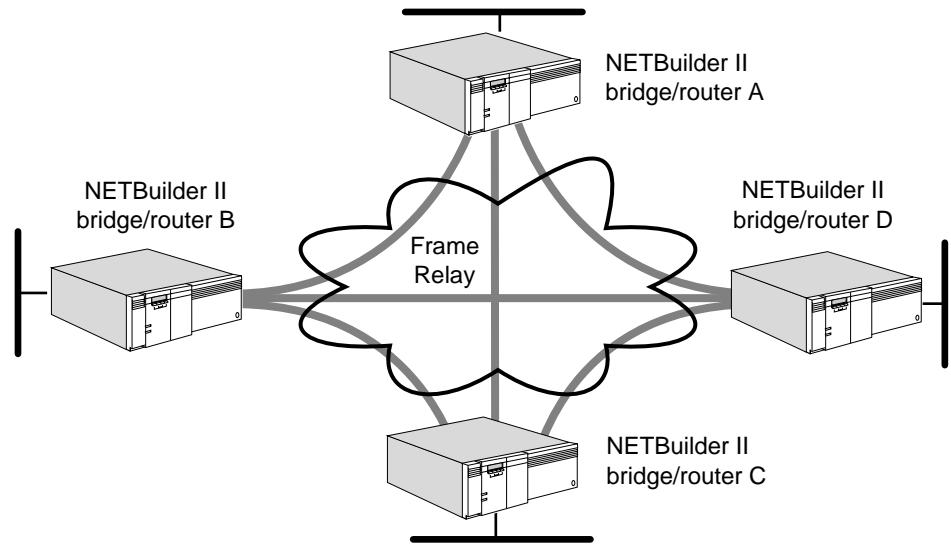
- In New York on an SVC assigned vcid 30. The -FR SvcLocalAddress parameter is being used to configure vcid 30 with telephone number 1-408-764-5734, and the -FR SvcDestAddress parameter is being used to configure the vcid 30 with telephone number 1-718-858-0859.
- In Boston on an SVC assigned vcid 35. This vcid has already been configured with a telephone number 1-408-764-6848 at the local site and 1-617-648-7652 at the remote site.
- In Atlanta on an SVC assigned vcid 40. This vcid has already been configured with a telephone number 1-408-764-8532 at the local site and 1-404-596-8598 at the remote site.
- In Miami on a PVC through DLCI 25. Since this DLCI is a PVC, a configuration with a telephone number is not required.

Figure 350 SVC and PVC Connections over Frame Relay



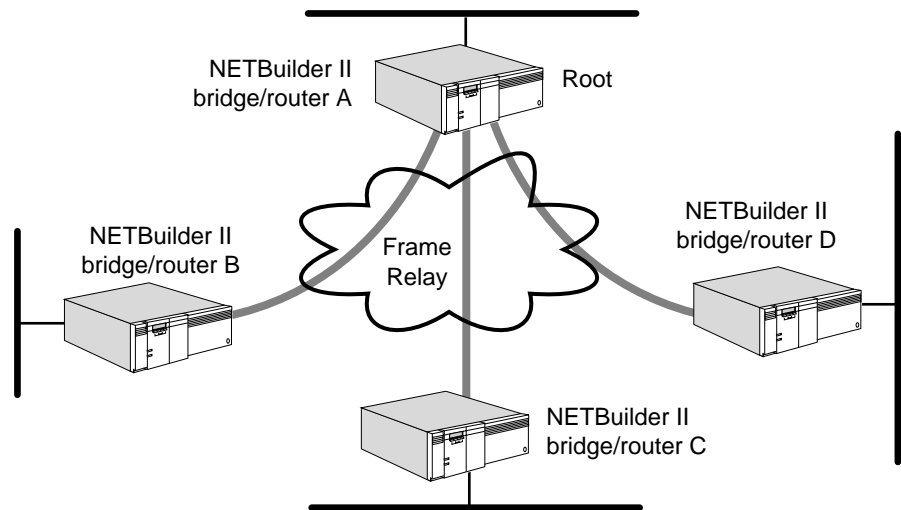
Fully Meshed, Partially Meshed, and Nonmeshed Topologies

A fully meshed Frame Relay topology (Figure 351) is a topology where each node on a network is directly connected to all other nodes on the network. Each node is connected to the other nodes through a PVC, and each PVC has a DLCI associated with it. This DLCI may appear as a different number to each end of the PVC.

Figure 351 Fully Meshed Frame Relay Topology

The topology in Figure 351 is composed of NETBuilder II bridge/routers. Through the established PVCs, bridge/router A is connected to bridge/routers B, C, and D; bridge/router B is connected to bridge/routers A, C, and D; and so on.

A nonmeshed Frame Relay topology (Figure 352) is a topology where each node on a network is not necessarily connected to all other nodes on the network.

Figure 352 Nonmeshed Frame Relay Topology

The topology in Figure 352 is composed of NETBuilder II bridge/routers. Through the established PVCs, bridge/router A is connected to bridge/routers B, C, and D. bridge/routers B, C, and D are connected to bridge/router A only, but not to one another.



Transparent bridging does not correctly operate in some nonmeshed topologies. For example, in Figure 353, the transparent bridge properly forwards traffic received on !v1 to !v2. However, traffic received from one of its remote connections on !v3 is not properly forwarded to the other two remote

connections on !v3; therefore, do not configure transparent bridging in this type of nonmeshed topology. The flooding algorithm floods packets on a per-port basis, not on a neighbor-per-port basis.

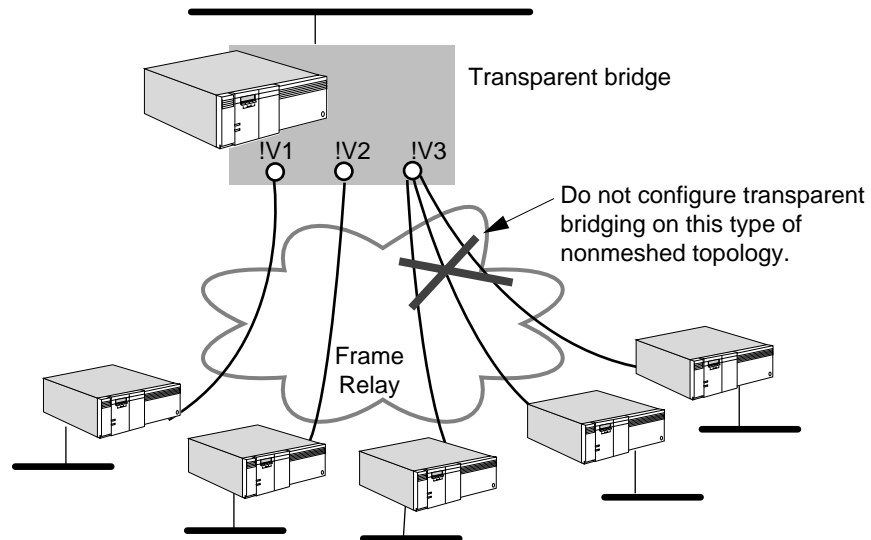


Figure 353 Transparent Bridging in Nonmeshed Frame Relay Topologies

A partially meshed Frame Relay topology is a topology where some nodes on a network are directly connected to nodes on the network (as in a fully meshed topology) and other nodes are not (as in a nonmeshed topology). Figure 354 is an example of a partially meshed Frame Relay topology.

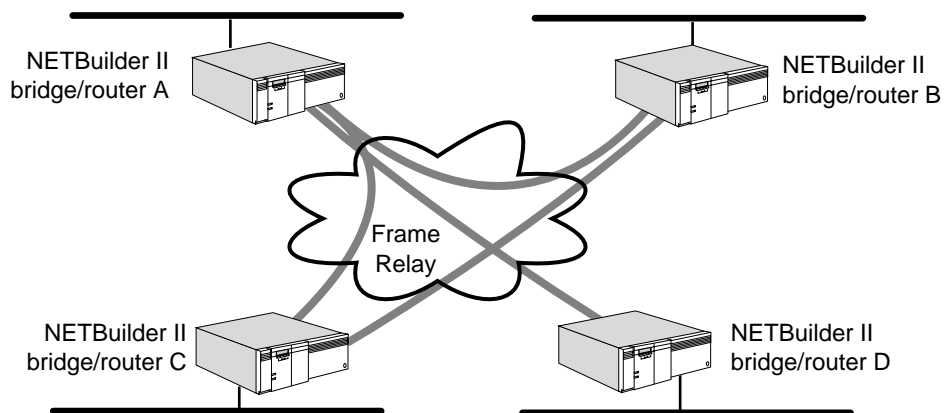


Figure 354 Partially Meshed Frame Relay Topology

The topology in Figure 354 is composed of four NETBuilder II bridge/routers. Through the established PVCs, bridge/routers A, B, and C are connected to one another, but bridge/router D is connected to bridge/router A only.

Two possible solutions exist to work around the lack of connectivity between bridge/routers B, C, and D in nonmeshed and partially meshed topologies. If you are routing IP-RIP, IPX, or AppleTalk, these protocols offer the next-hop split horizon feature. In IP-RIP, set `-RIP CONTROL` to `NonMesh` to enable next-hop split horizon. In IPX, next-hop split horizon is enabled by manually configuring neighbors. In AppleTalk, next-hop split horizon is enabled by adding static mappings to the address mapping table.

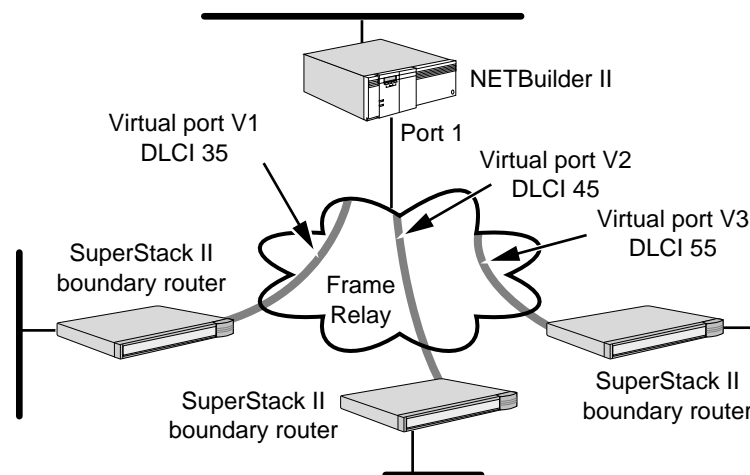
For example, if you are routing IP-RIP and you set `-RIP CONTROL` to `NonMesh`, a list of neighbors containing bridge/routers B, C, and D will be generated by the system (for more information, see *Reference for Enterprise OS Software*), or you can configure them as neighbors using the `-RIP AdvToNeighbor` parameter.

If routing IPX, you can configure bridge/routers B, C, and D as neighbors using the `-NRIP PolicyControl` and `-NRIP AdvToNeighbor` parameters. If routing AppleTalk, you can add the address of bridge/routers B, C, and D to an address mapping table. Bridge/router A, the root bridge/router, learns available routes from each neighbor and then updates each neighbor with available routes other than the routes of that particular neighbor. Even though bridge/routers B, C, and D are not directly connected to one another, they can still learn of routes other than their own through bridge/router A.

Another solution for the lack of connectivity is to create virtual ports. Virtual ports are supported by bridging and all routing protocols over a Frame Relay network. You must use virtual ports in a Boundary Routing over Frame Relay topology and when bridging or routing DECnet, VINES, or XNS over Frame Relay in a partially meshed or nonmeshed topology. Using virtual ports in all other bridging and routing scenarios over a Frame Relay network is optional. For information on the number of virtual ports supported per platform, see Table 11 in the Configuring Advanced Ports and Paths chapter.

Virtual ports allow the creation of multiple logical ports on one path. Each PVC attaches a separate logical network. Figure 355 is a Boundary Routing over Frame Relay topology where virtual ports are configured. In this topology, even though the SuperStack II boundary routers are not directly connected to one another, information about each of their networks can still be propagated through the NETBuilder II bridge/router.

Figure 355 Using Virtual Ports in a Boundary Routing Over Frame Relay Topology



For more information on virtual ports, see the Configuring Advanced Ports and Paths chapter. For more information on Boundary Routing over a Frame Relay topology, see the Configuring Boundary Routing System Architecture chapter.

Frame Relay Addresses

Before attaching your bridge/router to a Frame Relay network, obtain one or more virtual circuit identifiers, called DLCIs, from the Frame Relay service provider. A

DLCI identifies a circuit between two devices from the end users' perspective. Each end of the circuit can have a different DLCI number for the link.

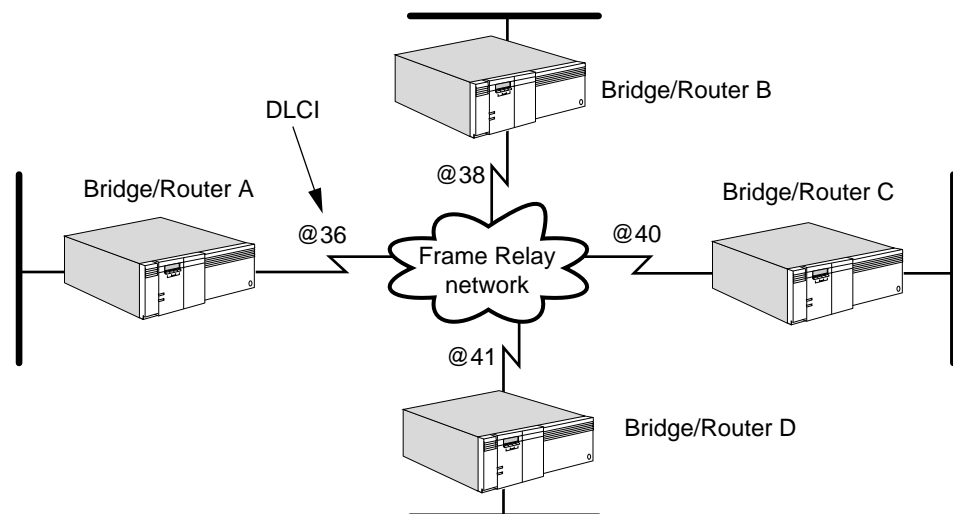
The DLCI number can range from 0 to 1023, but the service provider only assigns subscriber numbers ranging from 16 to 991. For ANSI and NTT LMI, 0 to 15 and 992 to 1023 are reserved. For LMI, 0 to 15 and 999 to 1023 are reserved.

In Figure 356, bridge/routers A, B, C, and D are assigned the DLCI numbers 36, 38, 40, and 41, respectively. The following items are examples of what occurs when packets are sent from one bridge/router to another:

- When bridge/router A sends a packet to bridge/router B, it uses DLCI 38. When the packet arrives at bridge/router B, the network changes the DLCI to 36 to indicate to bridge/router B that the packet originated at bridge/router A.
- When bridge/router A sends a packet to bridge/router C, it uses DLCI 40. When the packet arrives at bridge/router C, the network changes the DLCI to 36 to indicate to bridge/router C that the packet originated at bridge/router A.
- When bridge/router B sends a packet to bridge/router C, it also uses DLCI 40. When the packet arrives at bridge/router C, the Frame Relay network changes the DLCI to 38 to indicate to bridge/router C that the packet originated at bridge/router B.

3Com bridge/routers can operate in both local and global addressing schemes used by the Frame Relay network. In the standard (local) addressing convention, the DLCI number has only local significance; a duplicate number can be used by other bridge/routers. In the global addressing convention, identifiers used throughout the Frame Relay network are unique, and all traffic to a node has the same destination DLCI number.

Figure 356 Frame Relay Addressing Example



Local Management Interface Protocol

The LMI Protocol runs between the bridge/router data terminal equipment (DTE) and the Frame Relay network switching equipment data communications equipment (DCE). The LMI Protocol provides information about all devices that are accessible on the Frame Relay network by listing all DLCIs connecting the local system with the remote ones. The LMI Protocol improves reliability between the DTE and DCE by exchanging keepalive packets that are sent every 5 to 30

seconds, depending on the configuration. If the LMI Protocol is disabled, the bridge/router assumes that all the DLCIs are active whether they are up and running or not. The LMI Protocol is enabled by default on your bridge/router.

Some switches do not run the LMI Protocol. In this situation, set the -FR CONTROL parameter to NoLMI. For complete information on this parameter, see the FR Service Parameters chapter in *Reference for Enterprise OS Software*.

How Disaster Recovery Works

This section describes how disaster recovery works, including the use of virtual ports, dial-up, and leased lines.

Disaster recovery is a mechanism that allows you to maintain connectivity between your central and remote sites in the event of a primary line failure or the Frame Relay network failure. This section describes how to use virtual ports and explains possible points of failure in a Frame Relay network.



If you are configuring disaster recovery over Frame Relay on the peripheral node of a Boundary Routing environment, do not configure network resiliency or the central MAC address.

Using Virtual Ports for Disaster Recovery

To configure disaster recovery, you must create virtual ports. A PVC attached to the virtual port is designated as the primary PVC. After you create a virtual port, you can add a backup PVC to that virtual port to support disaster recovery. For additional information about configuring virtual ports over Frame Relay, see the Configuring Advanced Ports and Paths chapter.

You must configure virtual ports on both ends of the connection, and you must designate the same PVC on both ends of the connection as a primary PVC. When configuring a backup PVC, you must designate the same PVC on both ends of the connection as the backup PVC. Figure 348 is an example. The PVCs 5@30 at site A and 2@30 at site B are both designated as primary PVCs. 6@45 at site A and 3@45 at site B are both designated as backup PVCs. If you designate a PVC as primary on one end and backup on the other end, all packets are dropped.

You can configure the backup PVC on a separate link to provide redundancy. If the backup PVC is on a separate link, this link can be a leased line or a dial-up line. You can use a dial-up line only on one end of the connection. In most cases, a permanent connection is established between the router at the central site and the switch. At the remote site, the router establishes a connection with the switch using a dial-up link.

When using a dial-up link on a Frame Relay network, only the router can dial out by establishing a connection with the Frame Relay switch. The Frame Relay switch cannot dial out and can accept only incoming connections. When the switch accepts these incoming connections, it activates the PVC associated with that link. Since the switch cannot dial out to establish an end-to-end connection with the router, you must establish a permanent connection between the router and the Frame Relay switch on one side of your configuration. Establishing this connection enables the other side to dial out if the primary PVC fails.

The following points of failure are possible on a Frame Relay network:

- Failure of a local path
- Failure of a local switch

- Failure of a remote path
- Failure of a remote switch
- Failure in the Frame Relay network

A fully redundant network is one on which there is no single point of failure. In a partially redundant network, at least one point of possible failure exists. Data is not transmitted across a backup PVC when the primary PVC is active, which ensures correct sequencing of packets.

The triggering of the backup PVC is kept transparent to the network layer protocols by using DLCI substitution. In the case of a primary PVC failure, Frame Relay sends and receives data using the backup DLCI by substituting the primary DLCI for the backup DLCI in the packet before passing data to the network layer protocols. As long as a PVC exists that can carry the traffic on a virtual port, there is no change in port status, and the network layer protocols are unaware that the backup DLCI is being used.

In a Frame Relay disaster recovery environment, sessions may need to be restarted if a failure occurs on the primary PVC. By default LMI and Annex-D LMI provide full status information for the DLCIs every 60 seconds. If a failure occurs at one end of the primary PVC, it may take up to 60 seconds to inform the other end of the PVC. If a dial-up line is used for the backup PVC, additional time may be necessary to establish the connection. These delays can cause session timeout.



If you are configuring disaster recovery over Frame Relay on the peripheral node of a Boundary Routing environment, do not configure network resiliency. You must decide which type of redundancy best suits your needs.

Partially Redundant Networks

The following examples show the locations of redundant links and possible points of failure in partially redundant networks.

Example 1 In Figure 357, central site A is connected to remote site B. A redundant link (shown as the dotted line) is configured at site B, but not at site A. In this configuration, a failure of the link at site A or a failure of the Frame Relay network can bring down the connection between site A and site B.

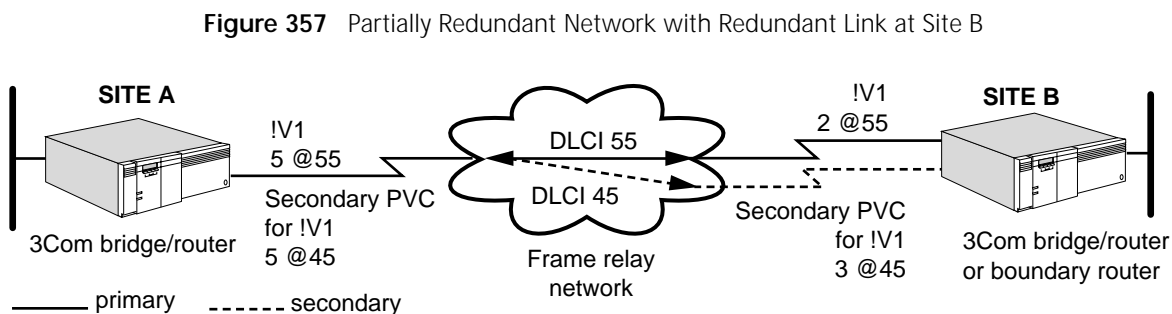


Table 87 shows the covered and uncovered links for Figure 357 if a primary line or switch fails.

Table 87 Covered and Uncovered Links

| Links Covered for Failure | Links Not Covered for Failure |
|---------------------------|-------------------------------|
| Link failure at site B | Link failure at site A |

Table 87 Covered and Uncovered Links

| Links Covered for Failure | Links Not Covered for Failure |
|---------------------------|-------------------------------|
| | Frame Relay network failure |

Example 2 Figure 358 shows a partially redundant configuration in which the redundant link is located at site A. At site B, a link failure or a Frame Relay network failure can bring down the connection between the two sites.

Figure 358 Partially Redundant Network with Redundant Link at Site A

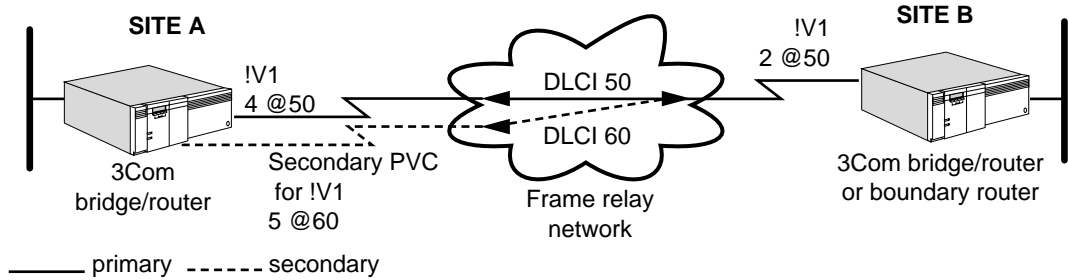


Table 88 shows the covered and uncovered links for Figure 358 if a primary line or switch fails.

Table 88 Covered and Uncovered Links

| Links Covered for Failure | Links Not Covered for Failure |
|---------------------------|-------------------------------|
| Link failure at site A | Link failure at site B |
| | Frame Relay network failure |

Example 3 In Figure 359, the network configuration has redundant links at both site A and site B. The point of failure in this configuration is the Frame Relay network.

Figure 359 Partially Redundant Network with Redundant Links at Site A and Site B

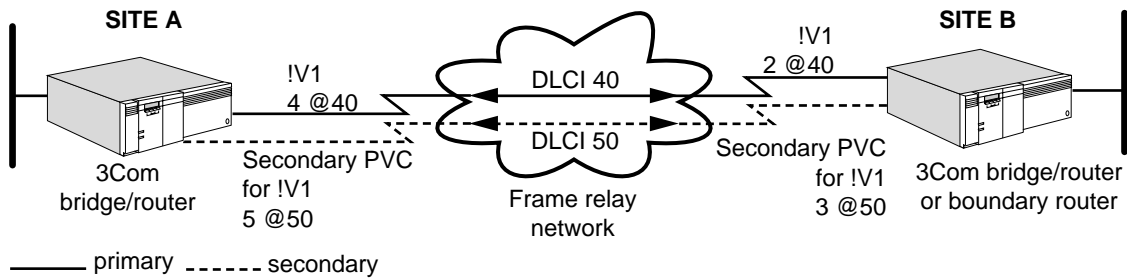


Table 89 shows the covered and uncovered links in this network configuration if a primary line or switch fails.

Table 89 Covered and Uncovered Links

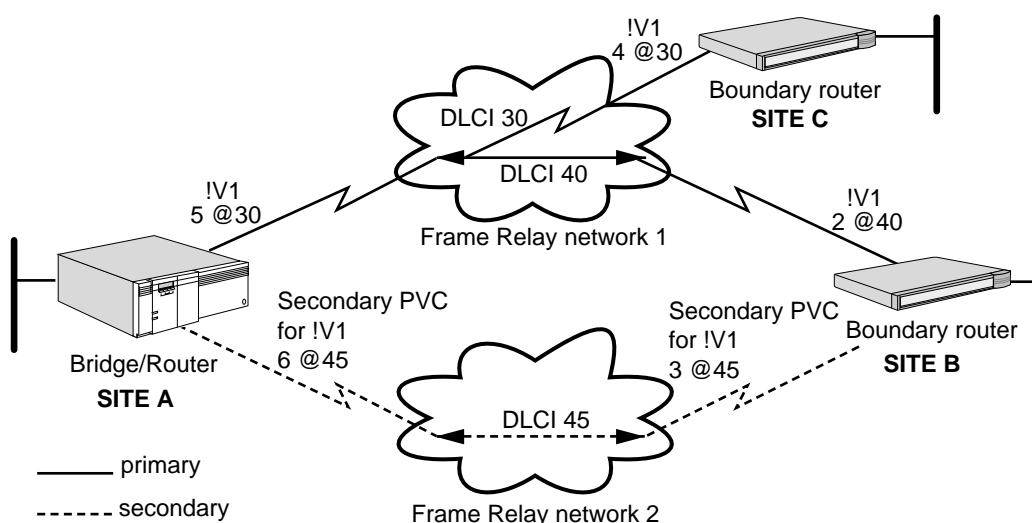
| Links Covered for Failure | Links Not Covered for Failure |
|----------------------------|-------------------------------|
| Link failure at either end | Frame Relay network failure |

Fully Redundant Networks

The following example shows the location of a redundant link between two sites.

Example 4 Figure 360 shows a Frame Relay configuration that is fully redundant between site A and site B. There is no single point of failure between sites A and B, because redundancy is provided for all possible points of failure between these nodes. Sites A and B have redundant links, and both sites A and B are connected to two different Frame Relay networks. DLCI 40 is the primary PVC going between site A and site B. DLCI 45 is the backup PVC between site A and site B. Each of these PVCs belongs to a different Frame Relay network accessible through different NETBuilder bridge/router interfaces. The primary and backup PVCs are on different interfaces. The network layer protocols believe they are talking to DLCI 40, even when the backup DLCI 45 has taken over. In this configuration, there is no redundancy between site A and site C.

Figure 360 Fully Redundant Network Between Site A and Site B



Frame Relay Congestion Control

Frame Relay network data traffic running over an individual virtual circuit of a NETBuilder bridge/router port will exceed network capacity at times. You can configure Frame Relay congestion control for individual virtual circuits to monitor and control the throughput of outgoing data when network capacity is exceeded.

You can configure two types of congestion control:

- One for NETBuilder bridge/router ports configured for NTLMI protocol
- One for NETBuilder bridge/router ports configured for other LMI protocols.

This section describes how congestion control for other LMI protocol users works. This section also describes how to avoid SNA performance problems when a port is configured for congestion control and LLC2 and SNA.

How Congestion Control Works

Congestion control, for bridge/router ports configured to use an LMI protocol other than NTLMI, optimizes the traffic throughput of a virtual circuit when the traffic exceeds the network capacity by having the bridge/router reduce its traffic load. After the congestion disappears, the bridge/router increases its traffic load again.

When a Frame Relay network is congested, it sets the frames with the Backward Explicit Congestion Notification (BECN) bit set to 1 to notify the NETBuilder

bridge/router of the congestion. When the first BECN=1 frame arrives at the bridge/router port to activate congestion control, the current throughput rate on the virtual circuit is reduced to a maximum rate equal to the value of cir.

To turn on congestion control and to set up how many consecutive BECN=1 frames must be sent before the current throughput rate on the virtual circuit is reduced to a maximum rate below the value of cir for the <vcid>, use:

```
ADD !<port> -FR CongestControl <vcid> [Yes | No] <step> (1-999)
```

<Vcid> identifies the Frame Relay virtual circuit to be set up for congestion control. The <vcid> for a PVC connection is the DLCI assigned by the Frame Relay service provider from a range of DLCI numbers. The <vcid> for an SVC connection is a DLCI that you assign from the range of DLCI numbers available that have not been assigned for a PVC connection.

<Step> specifies how many consecutive BECN=1 frames must be received before congestion control lowers the throughput rate to a value below cir. Zero is not allowed as a value for <step>, because it indicates that the no BECN=1 frames are required to activate congestion control. Also, <step> should not be set to too large a number because that large of an amount of BECN=1 frames may never reach the bridge/router port in time to activate congestion control while the traffic overload is occurring.

How the congestion control is executed is determined by the values entered for the -FR CIRbothdir parameter, which uses the following syntax:

```
ADD !<port> -FR CIRbothdir <vcid> <cir> <mincir> <Bc> <Be>
```

- <vcid> identifies the Frame Relay virtual circuit to be set up for congestion control. The <vcid> for a PVC connection is the DLCI assigned by the Frame Relay service provider from a range of DLCI numbers. The <vcid> for an SVC connection is a DLCI that you assign from the range of DLCI numbers available that have not been assigned for a PVC connection.
- <cir> (committed information rate) sets the maximum rate (in Kilobits per second) of NETBuilder bridge/router data that the network commits to transfer under normal conditions.
- <mincir> (minimum committed information rate) sets the minimum rate of NETBuilder bridge/router data throughput (in Kilobits per second) that the calling user is willing to accept for the call. If <cir> is set to 0, the value for <mincir> is used by the congestion control function.

For PVCs, enter for <mincir> the same value you enter for <cir>.

- <Bc> (committed burst size) sets the maximum of NETBuilder bridge/router data bits (in Kilobits) that the network commits to transfer under normal conditions during the time interval (Tc) measured in seconds.
- <Be> sets the maximum of uncommitted NETBuilder data bits (in Kilobits) that the network attempts to deliver in excess of <Bc> during the time interval (Tc) measured in seconds.

When configuring congestion control, follow these rules:

- Setting <step> to too large a number would mean that sufficient BECN=1 frames might not reach the bridge/router port in time to activate congestion control while the traffic overload is occurring.

- The <cir> and <Bc> values should be configured for a reasonable time interval (T_c) because $T_c = \langle Bc \rangle / \langle cir \rangle$. Typically, these values are configured for a T_c of one second.
- If the value of <Be> is set to nonzero, the NETBuilder bridge/router will be in a continuous monitoring throughput mode. The bridge/router will continuously monitor its throughput even if there are no BECN=1 frames coming in, which lowers the bridge/router performance.

When the first BECN=1 frame arrives at the bridge/router port to activate congestion control, the current throughput rate on the virtual circuit is reduced to a maximum rate equal to the value of cir. When the configured number of consecutive BECN=1 frames arrive at the bridge/router port, the maximum throughput rate lowers 1/4 to 0.750 times the value of cir as shown in Table 90.

Table 90 Congestion Control Throughput Rate Reduction

| Current NETBuilder Transmission Throughput Rate | Throughput Rate Reduction |
|---|---------------------------|
| cir > = Current Rate > (0.750) x cir | (0.750) x cir |
| (0.750) x cir > = Current Rate > (0.500) x cir | (0.500) x cir |
| (0.500) x cir > = Current Rate > (0.250) x cir | (0.250) x cir |
| (0.250) x cir > = Current Rate | No Action |

Further throughput rate reduction occurs only when the next number of consecutive BECN=1 frames arrive at the bridge/router port. After the reduction, a throughput increase of 1/8 the existing maximum throughput occurs when the port receives a number of consecutive BECN=0 frames equal to 1/2 of the number of consecutive BECN=1 frames configured for <step>.

For example, if you have configured <step> for 5 consecutive BECN=1 frames for a particular port, it requires 3 BECN=0 frames to increase the throughput by 1/8 of the existing maximum throughput because 2.5, which is half of 5, is rounded up to 3 frames.

The following example shows how congestion control reduces maximum throughput rates for a port as it receives BECN=1 frames, and increases maximum throughput rates as it receives BECN=0 frames.

In this example, cir is equal to 100 kbps, and step is 5. See Table 90 when going through this process:

- 1 The port receives one BECN=1 frame and the maximum transmit rate is 100 kbps or cir.
- 2 The port receives 5 consecutive BECN=1 frames, which limits the maximum transmit rate to 75 kbps.
- 3 The port receives 5 more consecutive BECN=1 frames, which limits the maximum transmit rate to 50 kbps.
- 4 The port receives 3 consecutive BECN=0 frames, which increases the maximum transmit rate to 50 times 1.125 = 56.25 kbps.
- 5 The port receives 3 more consecutive BECN=0 frames, which increases the maximum transmit rate to 56.25 times 1.125 = 63.28 kbps.
- 6 The port receives 5 more consecutive BECN=1 frames, which increases the limits to the maximum transmit rate equal to 50 kbps.

Frame Relay Congestion Control, LLC2, and SNA

Setting Frame Relay Congestion Control for a port that is configured for LLC2 and SNA over Frame Relay can lower the performance of SNA. The performance can decrease in an SNA network running many LLC2 sessions because LLC2 could take a long time in locating all existing LLC2 sessions to resume the transmission of SNA traffic over the Frame Relay network when it becomes uncongested. Poor response time is an indicator that the SNA performance could be affected by Frame Relay Congestion Control being active in the LLC2 protocol layer.

The -LLC2 FRCongestCont parameter is used to disable Frame Relay from sending messages when the network is congested and uncongested to the LLC2 layer. Disabling Frame Relay Congestion Control for LLC2 enables SNA to regain its former performance.

See “FRCongestCont” in the LLC2 Service Parameters chapter in *Reference for Enterprise OS Software* for details about this parameter.

Frame Relay Auto Startup

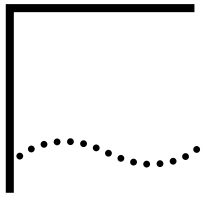
Auto startup does not determine the type of data network you subscribe to. After auto startup, you can set this value using the PDNtype parameter in the FR Service. This is a tuning feature.

For example, the following command configures port 2 to FR Service with the Sprint public data network:

```
SETDefault !2 -FR PDNtype = SPrint
```



This information applies to all NETBuilder bridge/router platforms.



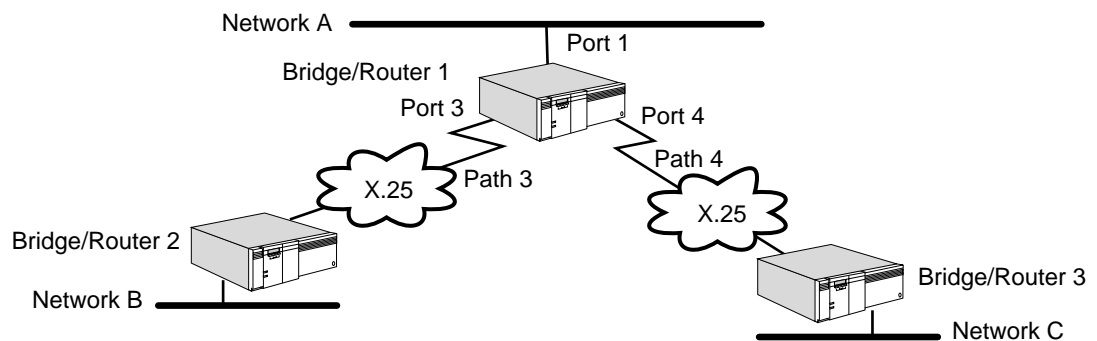
CONFIGURING WIDE AREA NETWORKING USING SMDS

This chapter describes how to configure your bridge/router to establish serial line connectivity through Switched Multimegabit Data Service (SMDS). It also describes how this wide area protocol works and it provides guidelines for operating, managing, and troubleshooting the protocol.

The wide area bridge/router supports bridging and routing over SMDS. SMDS allows your bridge/router to bridge or route over SMDS connectionless data service to other bridge/routers on the same wide area network.

SMDS, X.25, Asynchronous Transfer Mode (ATM), and Frame Relay allow only one path to be assigned to a port. Only one of these wide area protocols can run over a single path or serial line. For example, in Figure 361, SMDS is being run over port 3 on bridge/router 1, while X.25 is being run over port 4.

Figure 361 One Wide Area Protocol per Serial Line: SMDS and X.25



For conceptual information, see “How SMDS Works” later in this chapter.

Setting Up the SMDS Service

This section describes how to configure your bridge/router to transmit and receive data over an SMDS interface.

You must follow the steps in this section whether you are configuring for bridging or for routing. After you have completed these steps, proceed to “Setting Up Basic Bridging over SMDS” for bridging configuration information or to “Setting Up Basic Routing over SMDS” for routing configuration information.

For detailed descriptions of all commands and parameters, see *Reference for Enterprise OS Software*.

- Prerequisites** Before beginning this procedure, complete the following tasks:
- Log on to the bridge/router with Network Manager privilege.
 - Configure your wide area bridge/router ports and paths according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
 - Obtain SMDS individual and group addresses from your SMDS service provider. For more information, see “SMDS Addresses” later in this chapter.
 - If you need to connect the SMDS interface to more than 127 other routers, or to more than one logical network segment (or more than 32 logical segments under IP routing), or if you want to use selective filtering and route policies such as those described in “SMDS Addresses” later in this chapter, create virtual ports. For information about creating virtual ports, see “Configuring Virtual Ports” in the Configuring Advanced Ports and Paths chapter.

Procedure To allow your bridge/router to transmit and receive data over an SMDS network, follow these steps:

- 1 Assign an SMDS individual address for each port or virtual port to be used for SMDS Service, using:

```
SETDefault !<port> -SMDS SMDSIndivAddr = $C1<address>
```

SMDS individual addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the C in the address when reporting it to you, but you must include the C when configuring the bridge/router. The digit that follows the letter C is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

For packets received on the SMDS port, in addition to checking the address syntax, the software checks the first digit (country code). If the first digit is a 1, then the software flags the packet as an error if 10 digits do not follow the country code. This error appears as a syntactic error and can be displayed with the `SHoW -SYS STATISTICS -SMDS` command. For more information, see the Statistics Displays appendix. This address checking applies to both individual and group addresses.

- 2 Make sure that the bridge/router and the digital service units (DSUs) are configured identically.
 - a Enable the NewDXI option if it is supported by the DSU.

The NewDXI option corresponds to DXI 3.2 and is enabled by default. To verify this setting, enter:

```
SHoW -SMDS CONFIguration
```

If the setting is incorrect and the DSU supports DXI 3.2, change it by using:

```
SETDefault !<port> -SMDS CONTrol = NewDXI
```

A virtual port inherits its `CONTrol` value from the parent port. You cannot configure it directly.

- b Enable 32-bit cyclic redundancy check (CRC) on the path, if necessary.

Because some DSUs are configured for 32-bit CRC, the bridge/router must also be configured for the same value using:

```
SETDefault !<path> -PATH CONTrol = CRC32
```


- 3 Verify the clock, baud rate, and T1Mode settings for the path by entering:

SHow -PATH CONFIguration

The clock should be set to external, the baud rate should be set to 1,536 kbps, and the CONTrol parameter should be set to NoT1Mode (the default). If the settings are incorrect, change them using:

```
SETDefault !<path> -PATH CLock = External
SETDefault !<path> -PATH BAud = 1536
SETDefault !<path> -PATH CONTrol = NoT1Mode
```



If you change the clock or baud rate settings, you must re-enable the path before the new settings take effect, using SETDefault !<path> -PATH CONTrol = Enabled

- 4 If the DSU connected to the bridge/router is configured to use the Local Management Interface (LMI) Protocol, verify that LMI is enabled on the port or ports you are using for SMDS Service.

Confirm that the LMI Protocol is enabled using:

SHow [!<port>] -SMDS CONFIguration

The default is for LMI to be disabled. You can enable it using:

```
SETDefault !<port> -SMDS CONTrol = LMI
```

For information about the LMI Protocol, see “Local Management Interface Protocol” later in this chapter.

- 5 Enable the SMDS interface by setting the port owner to SMDS, using:

```
SETDefault !<port> -PORT OWNer = SMDS
```

This completes the procedure for configuring the SMDS Service.

Verifying the Configuration

To verify the SMDS configuration, enter:

SHow -SMDS CONFIguration

The bridge/router displays current SMDS configuration information. For information on using this parameter, see the SMDS Service Parameters chapter in *Reference for Enterprise OS Software*.

Setting Up Basic Bridging over SMDS

This section describes how to configure transparent and source route bridging over SMDS.

Configuring Transparent Bridging

This section describes how to configure your bridge/router for transparent bridging over the SMDS network.

Prerequisites

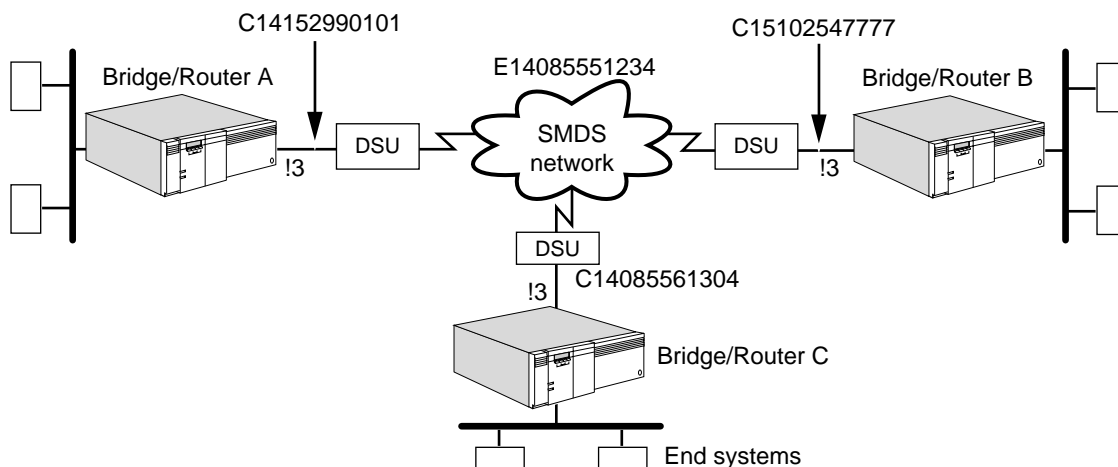
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in the Configuring Bridging chapter.
- Set up the SMDS Service as described in “How SMDS Works” later in this chapter.
- Contact your SMDS service provider and obtain group addresses. If you are using SMDS virtual ports, obtain a separate group address for each virtual port. For more information about group addresses, see “SMDS Addresses” later in this chapter.

Procedure

To enable transparent bridging to operate over the SMDS network based on the example in Figure 362, follow these steps.

Figure 362 Configuring Bridging over SMDS



- 1 Assign a group address to each port or virtual port of each bridge/router attached to the SMDS network using:

```
SETDefault !<port> -BRidge SMDSGroupAddr = $E1<address>
```

SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when configuring the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

On bridge/routers B and C, enter the same group address that you assigned to bridge/router A. The software uses this group address as a multicast address. When you transmit a packet from bridge/router A over the SMDS network using the group address, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- 2 Verify that transparent bridging has been enabled for the appropriate wide area port by entering:

```
SHow -BRidge CONFIguration
```

By default, bridging and transparent bridging are enabled on all ports.

If bridging has been disabled, enable it for the bridge/router by entering:

```
SETDefault -BRidge CONTrol = Bridge
```

If transparent bridging has been disabled for the wide area port (for example, on port 3, you can enable it by entering:

```
SETDefault !3 -BRidge TransparenTBRidge = TransparenTBRidge
```

Configuring Source Route Bridging

This section provides information for configuring source route bridging over SMDS.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in the Configuring Source Route Bridging chapter.
- Set up the SMDS Service as described in “How SMDS Works” later in this chapter.
- Contact your SMDS service provider and obtain group addresses. If you are using SMDS virtual ports, obtain a separate group address for each virtual port. For more information about group addresses, see “SMDS Addresses” later in this chapter.
- Assign a ring number to the SMDS wide area network.
- If your topology includes parallel bridges, determine unique bridge numbers.

Procedure

To configure source route bridging over SMDS, follow these steps:

- 1 Assign a group address to each port or virtual port of each bridge/router attached to the SMDS network using:

```
SETDefault !<port> -BRidge SMDSGroupAddr = $E1<address>
```

SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when you configure the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

Enter the same group address on all bridge/routers attached to the SMDS network. The software uses this group address as a broadcast address. When you transmit a packet from one bridge/router over the SMDS network using the group address, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- 2 Assign each wide area port of each bridge/router attached to the SMDS network the ring number of the network it accesses.

To assign a ring number, use:

```
SETDefault !<port> -SR RingNumber = <number> (1-4095) | 0x<number> (1-FFF)
```

You can enter the ring number in decimal or hexadecimal format. Precede a hexadecimal number with 0x.

- 3 Verify that source route bridging is enabled on the wide area port using:

```
SHow !<port> -SR CONFIguration
```

If source route bridging is disabled, you need to enable it for your wide area port. For example, to enable source route bridging on port 3, enter:

```
SETDefault !3 -SR SrcRouBridge = SrcRouBridge
```

- 4 Disable transparent bridging on the wide area port.

For example, to disable transparent bridging on port 3, enter:

```
SETDefault !3 -BRidge TransparentBRidge = NoTransparentBRidge
```

This step does not apply to model 32x and 52x SuperStack II NETBuilder bridge/routers. Transparent bridging is not supported on these bridge/routers.

- 5 Verify that bridging is enabled by entering:

```
SHow -BRidge CONFIguration
```

If bridging has been disabled, enable it for the bridge/router by entering:

```
SETDefault -BRidge CONTrol = Bridge
```

Setting Up Basic Routing over SMDS

This section describes how to configure your bridge/router to route data over an SMDS interface. The SMDS Service allows your bridge/router to perform routing over SMDS to other routers on the same wide area network.

Procedures for the following routing protocols are provided:

- AppleTalk
- DECnet
- IP
- IPX
- OSI
- VINES
- XNS

For detailed descriptions of all commands and parameters, see *Reference for Enterprise OS Software*.

Configuring AppleTalk

This section provides information for configuring AppleTalk routing with group addresses or individual addresses for communication over an SMDS network.

Prerequisites

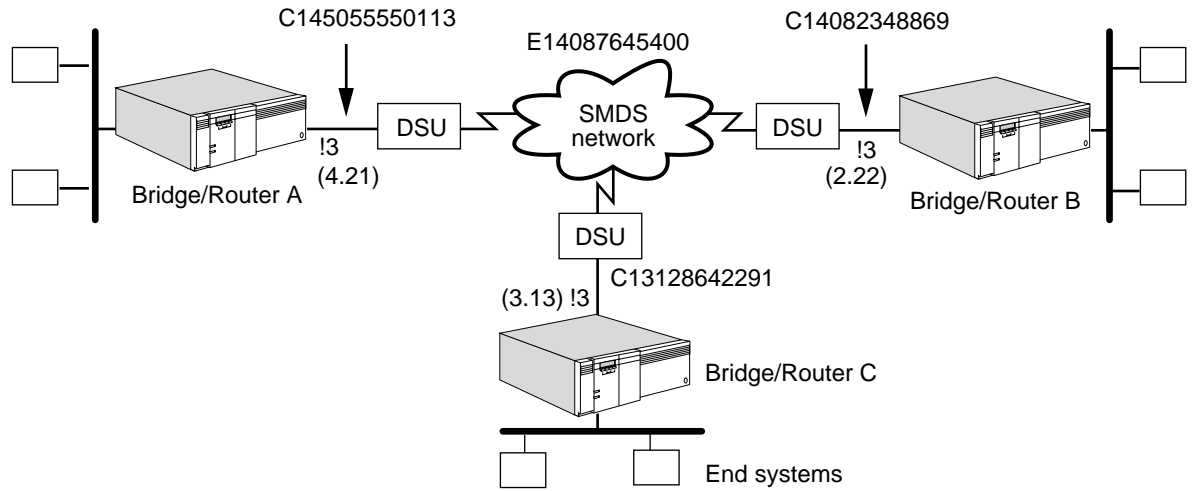
Before beginning this procedure, complete the following tasks:

- Configure your AppleTalk LAN according to the procedures in the Configuring AppleTalk Routing chapter.
- Set up the SMDS Service as described in "How SMDS Works" later in this chapter.
- Contact your SMDS service provider and obtain group addresses. If you are using SMDS virtual ports, obtain a separate group address for each virtual port. For more information about group addresses, see "SMDS Addresses" later in this chapter.
- Obtain the SMDS individual address of the remote router so that you can configure static mapping. For neighboring routers configured through static mapping, split horizon decisions are made at the next router link level. It allows for support of partially meshed and nonmeshed topologies. For neighboring routers configured through a group address, split horizon decisions are made at the port level.

Procedures

Use the following procedures and Figure 363 to enable the AppleTalk Protocol to operate over an SMDS network.

Figure 363 Configuring AppleTalk Routing over SMDS



Group Address Configuration

To configure AppleTalk routing over a SMDS network using a group address configuration, use Figure 363 as an example and follow these steps:

- 1 Assign a group address to each port or virtual port of each bridge/router attached to an SMDS network.

For example, to assign a group address to port 3 of bridge/router A in Figure 363, enter:

```
SETDefault !3 -AppleTalk SMDSGroupAddr = $E14087645400
```

SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when you configure the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

On bridge/routers B and C, enter the same group address that you assigned to bridge/router A. The software uses this group address as a multicast address. When you transmit a packet from bridge/router A over the SMDS network, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- 2 Assign a network number to the port or virtual port using:

```
SETDefault !3 -AppleTalk NetRange = <network-range>
```

- 3 Enable AppleTalk routing on the port attached to the SMDS network by entering:

```
SETDefault !3 -AppleTalk CONTROL = ROute
```

- 4 Assign a zone for the port attached to the SMDS network using:

```
ADD !3 -AppleTalk ZONE "<zone-string>" (1-32 char)
```

Individual Address Configuration

You can configure SMDS individual addresses for both non-AppleTalk and AppleTalk configurations.

Non-AppleTalk Configuration To configure AppleTalk routing over a SMDS network configured as a non-AppleTalk network, use Figure 363 as an example and follow these steps:

- 1 Configure all the ports on bridge/routers connected to the SMDS network to be connected to a non-AppleTalk network.

On bridge/routers A, B, and C, enter:

```
SETDefault !3 -AppleTalk CONTROL = NonAppleTalk
```

- 2 On each bridge/router, assign the SMDS individual address of the other bridge/routers ports and virtual ports connected to the network.

For example, on bridge/router A, enter:

```
ADD -AppleTalk ADDRESS !3 $C14082348869
```

```
ADD -AppleTalk ADDRESS !3 $C13128642291
```

Enter similar address information on bridge/routers B and C.

You can dynamically add and delete neighbors using the ADDRESS parameter while a port is enabled and AppleTalk is routing.

- 3 Enable routing on each AppleTalk bridge/router port attached to the SMDS network by entering:

```
SETDefault !3 -AppleTalk CONTROL = ROUTE
```

AppleTalk Configuration To configure AppleTalk routing over an SMDS network as an AppleTalk configuration, use Figure 363 as an example and follow these steps:

The example in the following procedure assumes that the network range for the SMDS cloud shared by the configured routers is 2 to 4 and that at least one of the routers is configured to send seed information to any other nonseed routers.

- 1 Specify the tentative network number and the tentative node ID for the specified port that the AppleTalk router uses during dynamic node address acquisition at port enable time using:

```
SETDefault !<port> -AppleTalk StartupNET = <number>(0-65279)
SETDefault !<port> -AppleTalk StartupNODE = <number>(0-253)
```

With these parameters, the local router can always assign the same AppleTalk node address to the local port, assuming that the address is within the network range assigned to the SMDS cloud. These static configurations are saved nonvolatile storage and only need to be changed when the topology changes.

- a For example, before routing is enabled on bridge/router A, enter:

```
SETDefault !3 -AppleTalk StartupNET = 4
SETDefault !3 -AppleTalk StartupNODE = 21
```

- b Enter values for the StartupNET and StartupNODE parameters for bridge/routers B and C.

- 2 Configure static mapping of SMDS individual addresses to their AppleTalk node addresses on each bridge/router's ports and virtual ports.

For example, on bridge/router A (AppleTalk address 4.21), enter the following SMDS individual addresses of the other routers connected to the SMDS network:

```
ADD -AppleTalk ADDRESS 2.22 $C14082348869
ADD -AppleTalk ADDRESS 3.13 $C13128642291
```

Configure static mapping of media addresses on bridge/routers B and C.

You can dynamically add and delete neighbors using the ADDRESS parameter.

- 3 Enable routing on each AppleTalk bridge/router port attached to the SMDS network by entering:

```
SETDefault !3 -AppleTalk CONTROL = ROUTE
```



To route through an SMDS network, you can either configure neighboring route through an SMDS group address, or configure using the -AppleTalk ADDRESS parameter, or you can configure both.

Configuring DECnet

This section provides information for configuring DECnet routing for communication over an SMDS network.

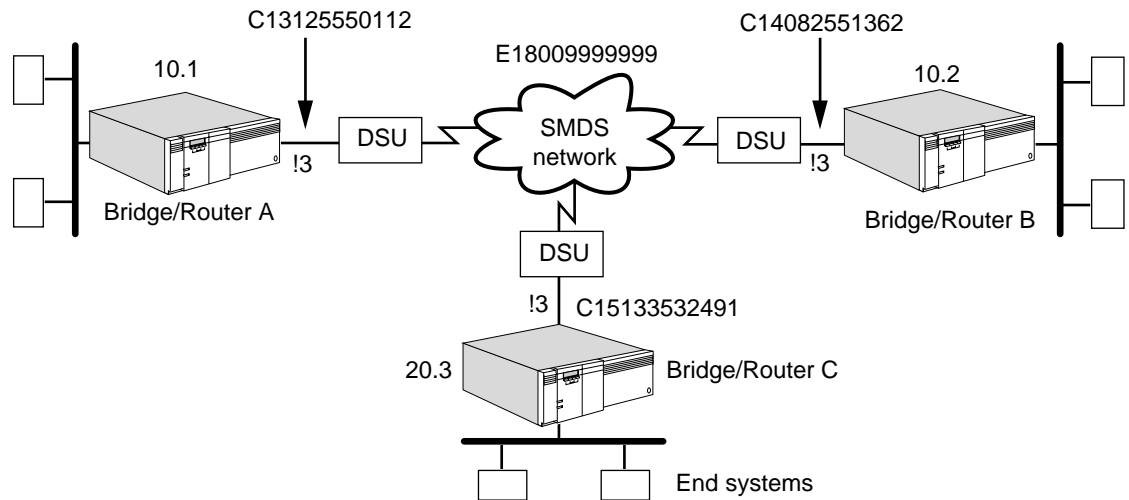
Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your DECnet LAN according to the procedures in the Configuring DECnet Routing chapter.
- Set up the SMDS Service as described in "How SMDS Works" later in this chapter.
- Contact your SMDS service provider and obtain group addresses. If you are using SMDS virtual ports, obtain a separate group address for each virtual port. For more information about group addresses, see "SMDS Addresses" later in this chapter.

Procedure

To enable the DECnet Protocol to operate over an SMDS network, use Figure 364 as an example and follow these steps:

Figure 364 Configuring DECnet Routing over SMDS

- 1 Assign a group address to each port or virtual port of each bridge/router attached to the SMDS network.

For example, to assign a group address to port 3 of bridge/router A in Figure 364, enter:

```
SETDefault !3 -DECnet SMDSGroupAddr = $E1800999999
```

SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when you configure the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

On bridge/routers B and C, enter the same group address that you assigned to bridge/router A. The software uses this group address as a multicast address for routing protocol packets. When you transmit a protocol packet from bridge/router A over the SMDS network, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- 2 Enable DECnet routing on each port or virtual port of each bridge/router attached to the SMDS network.

For example, to enable routing on port 3 of bridge/router A, enter:

```
SETDefault !3 -DECnet CONTROL = ROUTE
```

Enable routing on bridge/routers B and C.

Configuring IP

This section provides information for configuring IP routing for communication over an SMDS network.

Prerequisites

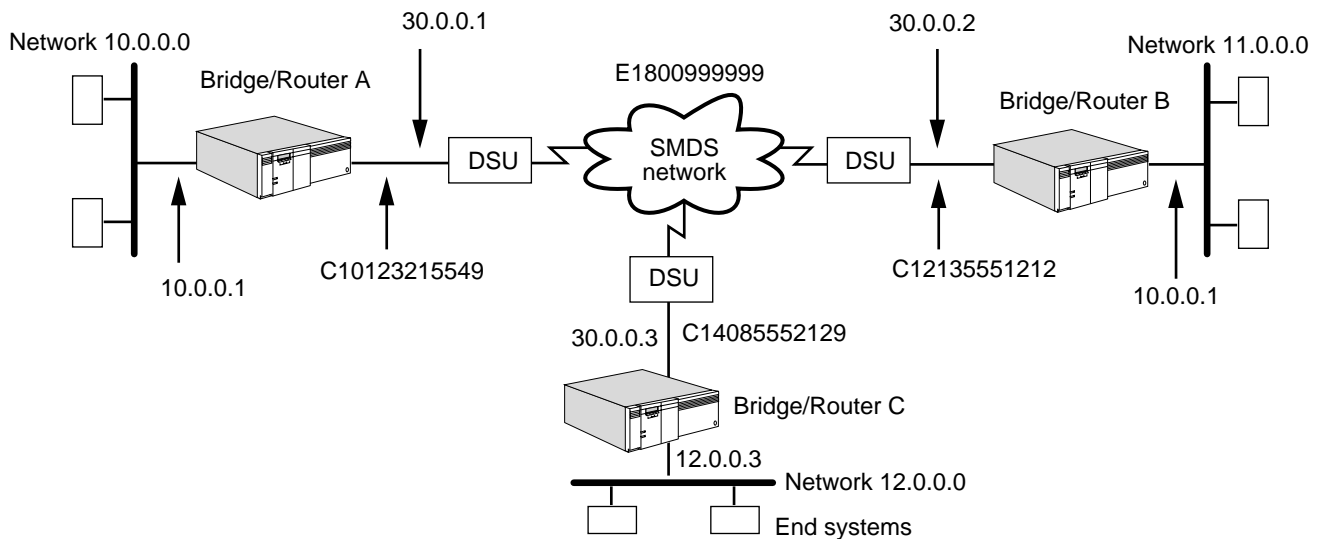
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in the Configuring IP Routing chapter.
- Set up the SMDS Service as described in “How SMDS Works” later in this chapter.
- Contact the SMDS Service provider, and obtain a group address. For more information about group addresses, see “SMDS Addresses” later in this chapter.
- Determine the IP addresses for each wide area port of each bridge/router attached to the SMDS network.

Procedure

To enable the IP Protocol to operate over an SMDS network, use Figure 365 as an example and follow these steps.

Figure 365 Configuring IP Routing over SMDS



- 1 Assign an IP address to each port or virtual port attached to the SMDS network. For example, the following command assigns the address 30.0.0.1 with subnet mask 255.255.255.0 to port 3 on bridge/router A:

```
SETDefault !3 -IP NETAddr = 30.0.0.1 255.255.255.0
```

Assign IP addresses for bridge/router B and C on the same subnet, for example 30.0.0.2, 30.0.0.3.

- 2 Specify the IP-to-SMDS group address mapping information per subnet.

For example, on each of bridge/routers A, B, and C, enter:

```
ADD -IP SMDSGroupAddr 30.0.0.0 $E1800999999
```

You may configure multiple IP subnets on the same SMDS port. If you do, you must specify IP address-to-SMDS group address mapping for each subnet.

SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when you configure the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

On bridge/routers B and C, enter the same group address that you assigned to bridge/router A. The software uses this group address as a multicast address for routing protocol packets. When you transmit a protocol packet from bridge/router A over the SMDS network, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- 3 Enable the dynamic routing protocols for IP using Routing Information Protocol for IP (RIP), Open Shortest Path First (OSPF), or Integrated IS-IS (IISIS).

- To learn routes dynamically on port 3 using RIP, enter:

```
SETDefault !3 -RIP CONTROL = (Talk, Listen)
```

Setting the CONTROL parameter to the TALK and Listen values enables the router to send and receive routing information with other routers using RIP.

- To enable routes dynamically on port 3 using OSPF, enter:

```
SETDefault !3 -OSPF CONTROL = Enable
```

Once OSPF operation is enabled, the router exchanges routing information with other routers using OSPF. OSPF does not support multiple IP subnets on a single SMDS port. Use virtual ports if you need multiple IP subnets on SMDS.

- To enable routes dynamically using Integrated IS-IS, see the Configuring IP Routing chapter.

- 4 Optionally, specify the network-to-router IP routing information to configure static routing.

In the example shown in Figure 365, the following sequence of commands uses the ADD -IP ROUTE <IP address> [<mask>] syntax to specify network-to-IP routing information for the bridge/routers and their respective networks directly attached to the SMDS wide area network.

On bridge/router A (IP address 30.0.0.1), enter:

```
ADD -IP ROUTe 11.0.0.0 30.0.0.2
```

```
ADD -IP ROUTe 12.0.0.0 30.0.0.3
```

Enter similar commands on bridge/router B (IP address 30.0.0.2) and bridge/router C (IP address 30.0.0.3), specifying the network-to-IP routing information.

5 Enable IP routing by entering:

```
SETDefault -IP CONTROL = ROUTe
```

This completes the procedure for configuring IP routing over SMDS.

Configuring IPX

This section provides information for configuring IPX routing for communication over an SMDS network.

Prerequisites

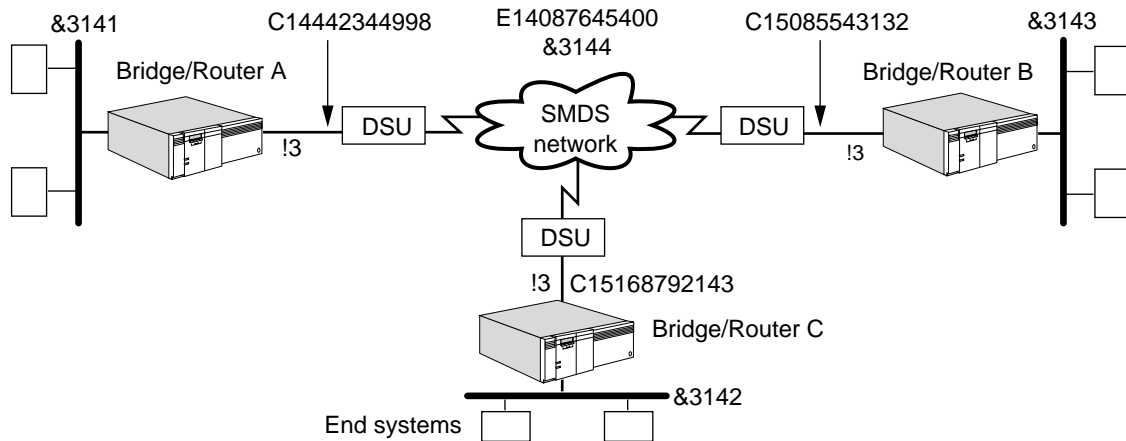
Before beginning this procedure, complete the following tasks:

- Configure your IPX LAN according to the procedures in the Configuring IPX Routing chapter.
- Set up the SMDS Service as described in "How SMDS Works" later in this chapter.
- Contact your SMDS service provider and obtain group addresses. If you are using SMDS virtual ports, obtain a separate group address for each virtual port. For more information about group addresses, see "SMDS Addresses" later in this chapter.
- Determine the IPX network number to be assigned to the bridge/routers.

Procedure

To enable the IPX Protocol to operate over an SMDS network, use Figure 366 as an example and follow these steps.

Figure 366 Configuring IPX Routing over SMDS



- 1 Assign a group address to each port or virtual port of each bridge/router attached to the SMDS network.

For example, to assign a group address to port 3 of bridge/router A in Figure 366, enter:

```
SETDefault !3 -IPX SMDGroupAddr = $E14087645400
```

SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when you configure the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

On bridge/routers B and C, enter the same group address that you assigned to bridge/router A. The software uses this group address as a multicast address. When you transmit a packet from bridge/router A over the SMDS network, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- 2 Assign a network number to each port or virtual port attached to the SMDS network.

For example, to assign a network number to port 3 of bridge/router A, enter:

```
SETDefault !3 -IPX NETnumber = &3144
```

Assign the same network number to bridge/routers B and C.

- 3 Verify that IPX routing is enabled on each bridge/router attached to the SMDS network by entering:

```
SHow -IPX CONFIguration
```

If routing has been disabled on the SMDS port of bridge/router A, enable it by entering:

```
SETDefault !3 -IPX CONTrol = ROute
```

Enable routing on bridge/routers B and C.

- 4 Verify that dynamic learning is enabled on each wide area port of each bridge/router attached to the SMDS network.

The -NRIP CONTrol and -SAP CONTrol parameters are set to TALK and Listen by default. To verify this setting for bridge/router A, enter:

```
SHow !3 -NRIP CONTrol
```

```
SHow !3 -SAP CONTrol
```

If the settings are not correct, you need to change the settings. For example, to enable dynamic learning on port 3 of bridge/router A, enter:

```
SETDefault !3 -NRIP CONTrol = (TAlk, LIsten)
```

```
SETDefault !3 -SAP CONTrol = (TAlk, LIsten)
```

Verify the settings on bridge/routers B and C.

- 5 Configure an internal network number on WAN links where only routers are attached using:

```
SETDefault -IPX InternalNET = &<number>(1-FFFFFFFD)
```

- 6 Enable the NetWare Link Services Protocol (NLSP) Protocol on the WAN links and disable NetWare Routing Information Protocol (NRIP) and Services Advertising Protocol (SAP) by entering:

```
SETDefault !3 -NLSP CONTROL = Enable
SETDefault !3 -NRIP CONTROL = (NoTalk, NoListen)
SETDefault !3 -SAP CONTROL = (NoTalk, NoListen)
```

By disabling NRIP and SAP, you conserve network bandwidth which is useful over WAN links. The NLSP Protocol uses the SMDS group address to send and receive routing packets.

- 7 Display the NLSP adjacencies by entering:

```
SHow -NLSP ADJAcencies
```

Configuring OSI This section provides information for configuring OSI routing for communication over an SMDS network.

Prerequisites

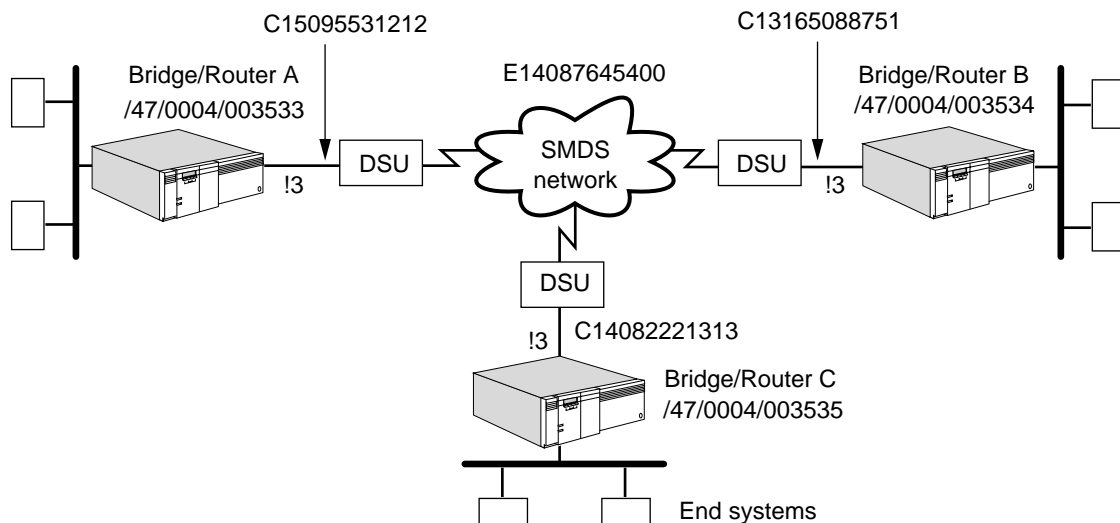
Before beginning this procedure, complete the following tasks:

- Configure your OSI LAN according to the procedures in the Configuring OSI Routing chapter.
- Set up the SMDS Service as described in “How SMDS Works” later in this chapter.
- Contact your SMDS service provider and obtain group addresses. If you are using SMDS virtual ports, obtain a separate group address for each virtual port. For more information about group addresses, see “SMDS Addresses” later in this chapter.

Procedure

To enable the OSI Protocol to operate over an SMDS network, use Figure 367 as an example and follow these steps.

Figure 367 Configuring OSI Routing over SMDS



- 1 Assign a group address to each port or virtual port of each bridge/router attached to the SMDS network.

For example, to assign a group address to port 3 of bridge/router A in Figure 367, enter:

```
SETDefault !3 -ISIS SMDSGroupAddr = $E14087645400
```

SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when you configure the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

On bridge/routers B and C, enter the same group address that you assigned to bridge/router A. The software uses this group address as a multicast address. When you transmit a packet from bridge/router A over the SMDS network, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- 2 Verify that IS-IS routing is enabled on each bridge/router attached to the SMDS network by entering:

```
SHow -CLNP CONFIguration
```

If routing has been disabled, enable it on bridge/router A by entering:

```
SETDefault -CLNP CONTrol = Route
```

Enable routing on bridge/routers B and C.

Configuring VINES This section provides information for configuring VINES routing for communication over an SMDS network.

Prerequisites

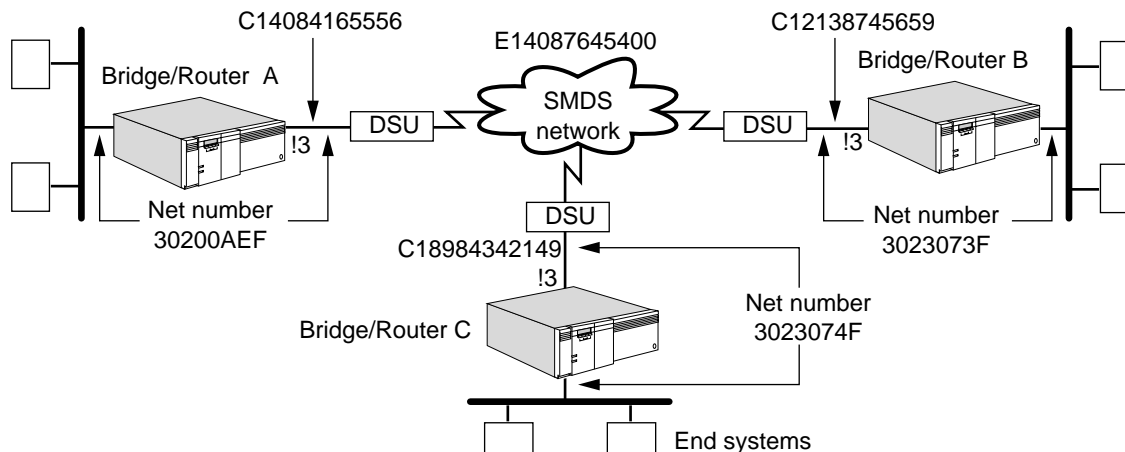
Before beginning this procedure, complete the following tasks:

- Configure your VINES LAN according to the procedures in the Configuring VINES Routing chapter.
- Set up the SMDS Service as described in "How SMDS Works" later in this chapter.
- Contact your SMDS service provider and obtain group addresses. If you are using SMDS virtual ports, obtain a separate group address for each virtual port. For more information about group addresses, see "SMDS Addresses" later in this chapter.

Procedure

To enable the VINES Internet Protocol (VIP) to operate over an SMDS network, use Figure 368 as an example and follow these steps.

Figure 368 Configuring VINES Routing over SMDS



- 1 Assign a group address to each port or virtual port of each bridge/router attached to the SMDS network.

For example, to assign a group address to port 3 of bridge/router A in Figure 368, enter:

```
SETDefault !3 -VIP SMDSGroupAddr = $E14087645400
```

- 2 SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when you configure the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

On bridge/routers B and C, enter the same group address that you assigned to bridge/router A. The software uses this group address as a broadcast address. When you transmit a packet from bridge/router A over the SMDS network, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- Verify that VINES routing is enabled on each bridge/router attached to the SMDS network by entering:

```
SHoW -VIP CONFIguration
```

If routing has been disabled, enable it on bridge/router A by entering:

```
SETDefault !3 -VIP CONTrol = Route
```

- Enable routing on bridge/routers B and C.

Configuring XNS

This section provides information for configuring XNS routing for communication over an SMDS network.

Prerequisites

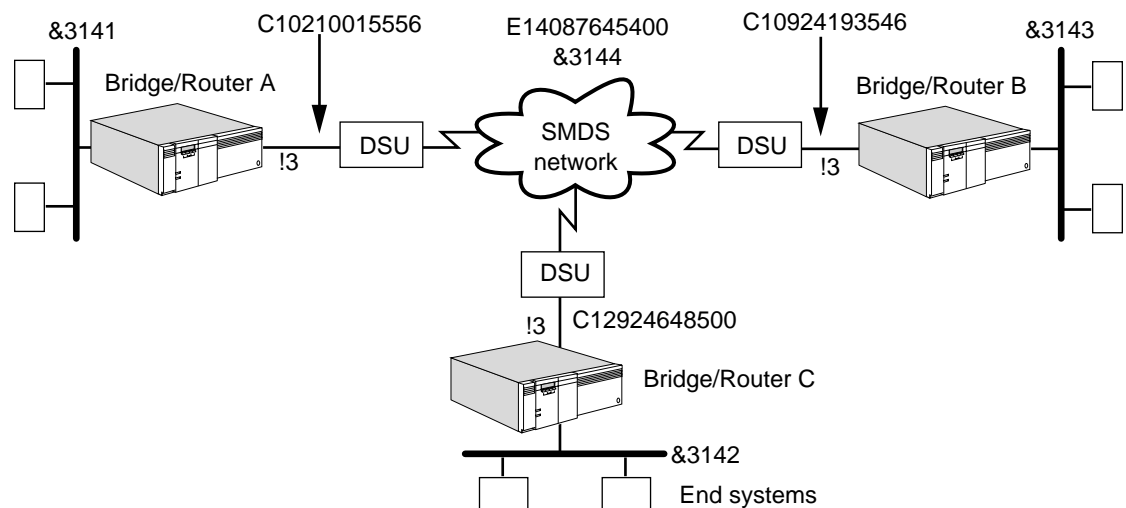
Before beginning this procedure, complete the following tasks:

- Configure your XNS LAN according to the procedures in the Configuring XNS Routing chapter.
- Set up the SMDS Service as described in "How SMDS Works" later in this chapter.
- Contact your SMDS service provider and obtain group addresses. If you are using SMDS virtual ports, obtain a separate group address for each virtual port. For more information about group addresses, see "SMDS Addresses" later in this chapter.
- Determine the XNS network number to be assigned to the bridge/routers.

Procedure

To enable the XNS Protocol to operate over an SMDS network, use Figure 369 as an example and follow these steps:

Figure 369 Configuring XNS Routing over SMDS



- 1 Assign a group address to each port or virtual port of each bridge/router attached to the SMDS network.

For example, to assign a group address to port 3 of bridge/router A in Figure 369, enter:

```
SETDefault !3 -IDP SMDSGroupAddr = $E14087645400
```

SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when you configure the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. You should use the country code for your own country reported by your SMDS service provider.

On bridge/routers B and C, enter the same group address that you assigned to bridge/router A. The software uses this group address as a multicast address. When you transmit a packet from bridge/router A over the SMDS network, all bridge/routers in the same group receive the packet.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, because Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

- 2 Assign a network number to each port or virtual port of each bridge/router attached to the SMDS network.

For example, to assign a network number to port 3 of bridge/router A, enter:

```
SETDefault !3 -IDP NETnumber = &3144
```

Assign the same network number to bridge/router B and bridge/router C.

- 3 Verify that dynamic learning is enabled on each port of each bridge/router attached to the SMDS network.

The CONTROL parameter is enabled by default. To verify this setting for bridge/router A, enter:

```
SHow !3 -RIPXNS CONTrol
```

If the CONTROL parameter is not enabled, you need to enable it. For example, to enable it on port 3, enter:

```
SETDefault !3 -RIPXNS CONTrol = Enabled
```

Verify that dynamic learning is enable on bridge/routers B and C.

- 4 Verify that IDP routing is enabled on each bridge/router attached to the SMDS network by entering:

```
SHow -IDP CONFIguration
```

If IDP routing has been disabled, enable it on bridge/router A by entering:

```
SETDefault -IDP CONTrol = Route
```

Enable routing on bridge/routers B and C.

How SMDS Works

SMDS is a connectionless packet switched service provided by telephone companies. Hosts or internetworking equipment connected to SMDS nodes can exchange packets across the wide area network.

To connect a router to an SMDS network, three levels of the SMDS Interface Protocol (SIP) must be supported. Your 3Com router provides the SIP-3 Protocol,

which encapsulates the user data into an L3PDU. SIP-1 and SIP-2 are provided by a third-party CSU/DSU.

The sections that follow provide some basic information about SMDS Service:

- SMDS addresses
- LMI Protocol

SMDS Addresses

SMDS addresses are of two types: individual addresses, for unicast traffic, and group addresses, for multicast traffic. The addresses are distinguished by the value of the first or control digit, which has the value hexadecimal C for an individual address and hexadecimal E for a group address. Each address has 15 decimal digits following the control digit, and resembles a telephone number. If an address has fewer than 15 digits, the software automatically right-pads it with hexadecimal Fs to the full length.

| | |
|------------------|--------------------|
| C14085551212FFFF | Individual Address |
| E14085551234FFFF | Group Address |

An individual address routes data to a unique node, a device attached to an SMDS network through a Subscriber Network Interface (SNI). The SMDS service provider assigns a block of up to 16 individual addresses to each SNI. Enterprise OS software can use the extra addresses to create virtual SMDS ports through the SNI. For information about configuring virtual ports on SMDS, see “Configuring Virtual Ports” in the Configuring Advanced Ports and Paths chapter.

SMDS permits multiple nodes to be assigned the same group address (in addition to their individual addresses). Packets sent to a group address are delivered to all nodes in the group. This feature gives SMDS the appearance of a LAN.

Local Management Interface Protocol

The LMI Protocol runs between the bridge/router data terminal equipment (DTE) and the CSU/DSU data communications equipment (DCE). The LMI Protocol improves reliability between the DTE and DCE by exchanging heartbeat packets every 5 to 30 seconds, depending on the configuration.

If the LMI Protocol is not enabled, the line between the router and the CSU/DSU is assumed to be up. The LMI Protocol is disabled by default on your bridge/router.



Some DSUs do not run the LMI Protocol. In this case, set the CONTROL parameter in the SMDS Service to NoLMI (the default setting).

SMDS Service Limits

The SMDS Service sets upper limits on the number of members in a group, the number of groups an individual address can belong to, and the total number of addresses (individual and group) that any one SNI can exchange packets with.

- Each group address can represent up to 128 individual addresses.
- Each individual address can belong to up to 32 groups.
- A single SNI can exchange data among 128 total individual or group addresses.

The set of addresses that an SNI can exchange data with is configured by the service provider, following the subscriber's specifications, into a feature of the SMDS switch called the address screen. The NETBuilder bridge/router does not implement the address screen and is not aware of it.



Although an SMDS virtual port can have more than one group address, a group address cannot be shared by multiple ports on the same NETBuilder bridge/router, since Enterprise OS software uses the group address to identify the virtual port for which a packet is intended.

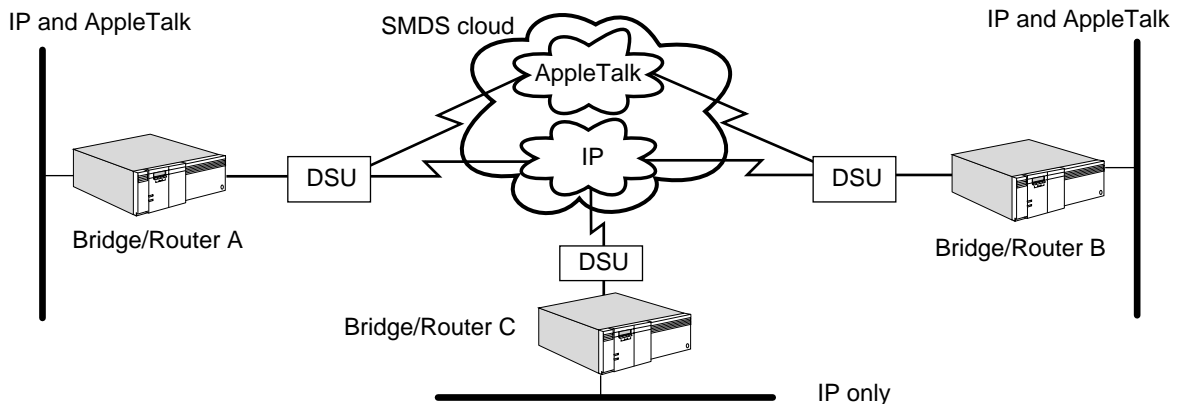
SMDS group addresses can be used in a variety of applications where it is desirable to divide nodes on the network into several groups that are treated in different ways. The following sections give some examples of these applications.

Separating Routing Protocols

The simplest SMDS configuration allows each router to exchange data with each of the subscriber's other routers, creating a full mesh across the SMDS network. Within this configuration, group addresses can be used to separate routing protocols. For example, all routers support IP, so all routers would belong to the IP group. Only routers that support AppleTalk would belong to the AppleTalk group. By addressing them to the AppleTalk group, AppleTalk routing updates and name service queries can be sent only to AppleTalk routers.

Figure 370 illustrates this configuration. Routers A and B route both IP and AppleTalk. Router C routes only IP. Routers A and B are assigned one SMDS group address (creating an AppleTalk group), while all three routers are assigned another SMDS group address (creating an IP group). When the routing protocols have been properly set up, AppleTalk routing and name service broadcast packets are delivered only to routers A and B, not to router C.

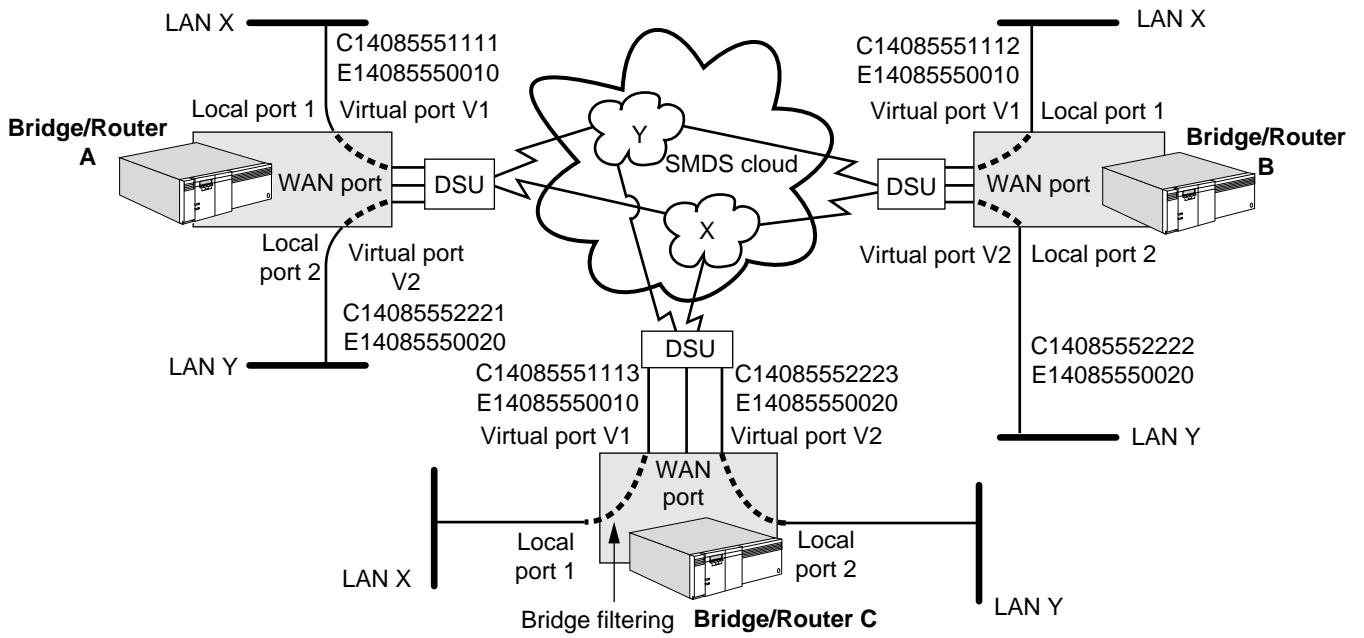
Figure 370 SMDS Full Mesh Configuration with Two Groups



Transparent Bridging

A more complex configuration might use virtual ports to provide additional control over the traffic. Consider a situation in which transparent bridging over SMDS is configured. Several organizations whose LANs are located close together (for example, several small companies in the same office building) all need a wide area connection to their branch offices. These organizations would like to share the cost of a bridge, but they do not want to compromise the privacy of their data or allow others to use bandwidth that they pay for. Virtual ports over SMDS, together with bridge filtering, can allow these organizations to share equipment without mixing bandwidth or broadcast traffic. The traffic of each organization is filtered to a separate virtual port, and the SMDS group address is used to identify these virtual ports.

Figure 371 Transparent Bridging over SMDS



In Figure 371, LANs X and Y share bridge/routers A, B, and C, one at each location, which are all configured as bridges. Each bridge has two local ports. On each bridge, LAN X is attached to local port 1, and LAN Y is attached to local port 2.



In practice, LANs X and Y may be close to each other at only one location, not three. The techniques described in this section can be used to separate traffic at that location.

Each bridge also has a wide area port, which has been configured for the SMDS Service, as described in this chapter. Enterprise OS software has also been used to create two virtual ports, V1 and V2, for this SMDS port, again on all three bridges. (One of these two ports could actually be the parent port rather than a virtual port.) At each bridge, the parent SMDS port is used to configure the SMDS CONTrol parameter, selecting the data exchange interface (DXI) that matches the DSU, and enabling or disabling LMI operation.

Two SMDS group addresses, E14085550010 and E14085550020, have been obtained from the SMDS service provider. Group address E14085550010 is assigned to virtual port V1 on all three bridges, and group address E14085550020 is assigned to virtual port V2 on all three bridges, as described in this chapter. Each virtual port on each bridge also has a unique SMDS individual address, as required by the SMDS Service.

The bridge filters on each bridge are configured so that packets are bridged only between virtual SMDS port V1 and local port 1, and between virtual SMDS port V2 and local port 2.

The bridge filters can be configured using the following commands. First, set the default action of the Filter Service to Discard by entering:

```
SETDefault -Filter DefaultAction = Discard
```

Define a filter mask called ANY that matches any packet by entering:

```
ADD -Filter MASK ANY %0 | %ff = %ff
```

Add filter policies using the mask ANY by entering:

```
ADD -Filter POLicy LANX-V1 forward ANY between !1 and !V1  
ADD -Filter POLicy LANY-V2 forward ANY between !2 and !V2
```

At each bridge, traffic from LAN X travels over local port 1 and is bridged to virtual SMDS port V1, where it is multicast to group address E14085550010. Traffic from LAN Y travels over local port 2 and is bridged to virtual SMDS port V2, where it is multicast to group address E14085550020.

The SNI address screen is configured as a full mesh, so all SMDS traffic from each bridge is sent to the other two bridges. At each bridge, traffic received for group address E14085550010 is assigned to virtual port V1 and bridged to local port 1, which is attached to LAN X. Traffic received for group address E14085550020 is assigned to virtual port V2 and bridged to local port 2, which is attached to LAN Y.

Even local bridging between ports attached to the same bridge is filtered, so data from the two organizations is always kept separate.

- Source Route and Transparent Bridge Separation** You may require source route bridging over the SMDS cloud between some LAN ports (for instance, token ring and FDDI) but not others. To keep source-route-bridged traffic separate from transparently bridged traffic, you can create a virtual SMDS port to carry one kind of bridged traffic, and use the parent port or another virtual port for the other.
- AppleTalk Route Filtering** Route filtering in AppleTalk is configured for each port with of the NetFilter parameter. You can selectively filter routing information learned on one port and propagated to other ports by creating virtual SMDS ports and distinct SMDS groups. Entity filtering in AppleTalk is controlled in a similar way by the EntityFilter and EntityFilterNum parameters and can be propagated selectively by the same technique.
- IPX Migration from RIP/SAP to NLSP** Over IPX routing, SMDS virtual ports can be used for phased introduction of NLSP to the network, where some remote bridge/routers have not yet been upgraded to support NLSP but still support RIP/SAP. Instead of defaulting to RIP/SAP, those remote bridge/routers that understand NLSP can be collected into a new subgroup, while RIP/SAP routers remain in the original subgroup until they can be upgraded.
- IP Route Policy** With IP routing, you can use SMDS virtual ports to control routing information with varying policies or protocols among the different SMDS virtual ports. For instance, one subgroup of equipment may already be using OSI IS-IS to support CLNP. The solution is to enable Integrated IS-IS selectively for these nodes under IP.

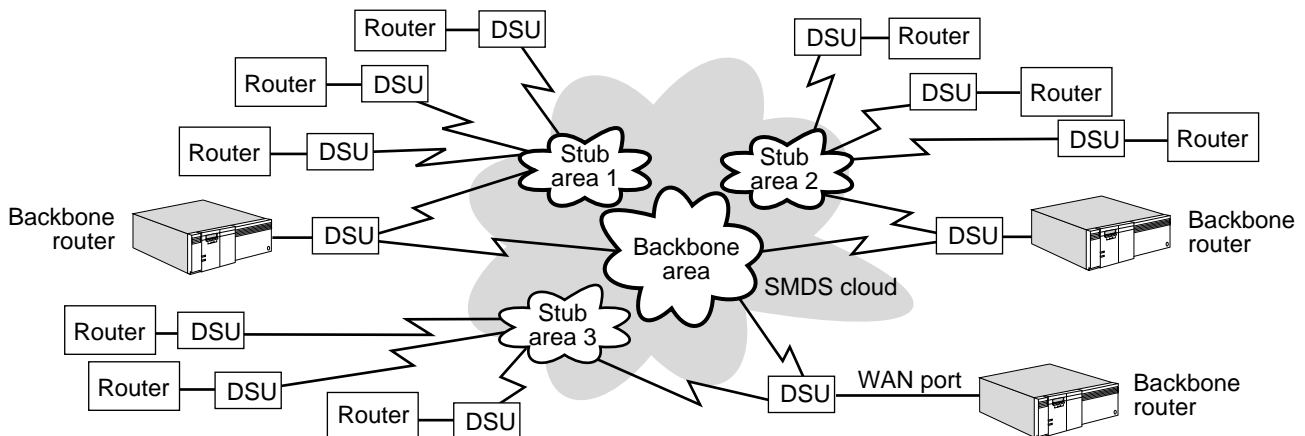
Large Hierarchical Networks

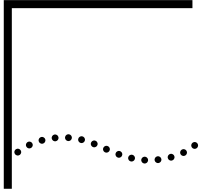
You can connect a large IP network over an SMDS cloud by combining the multiple area techniques of OSPF with SMDS virtual ports. This hierarchical approach expands the total number of bridge/routers that can be interconnected over SMDS by limiting the number that must communicate directly. Dividing the SMDS-connected bridge/routers into regions has two advantages:

- The SMDS address screen limitations are bypassed because each backbone router need communicate only with its own stub area and the other backbone routers. Different stub areas do not need to belong to the same address screen; they communicate through the backbone.
- The size of the OSPF database is reduced, saving network bandwidth for data.

The network bandwidth and router CPU time saved by OSPF summarization techniques will, in many cases, compensate for the extra hop needed by traffic traveling from a stub area to the backbone or another stub area. This configuration also saves the cost of additional SNIs that would otherwise be needed for the regional and backbone routers.

Figure 372 Large Hierarchical SMDS Network



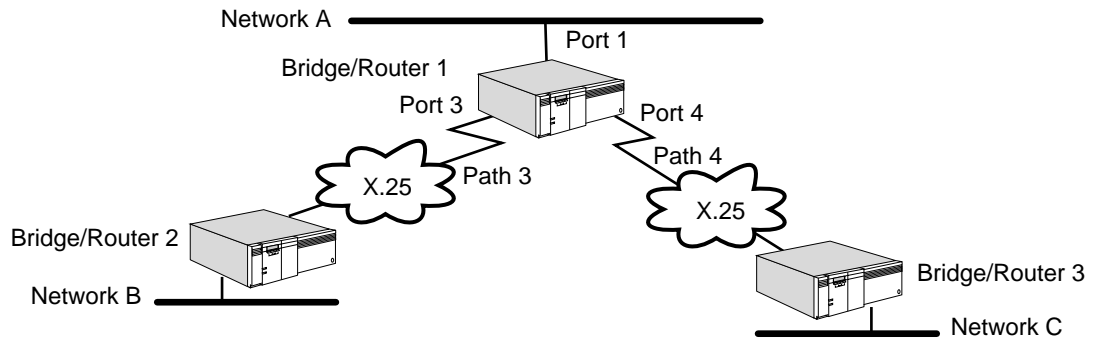


CONFIGURING WIDE AREA NETWORKING USING X.25

This chapter describes the procedures for preparing your wide area bridge/router for X.25 wide area networking and describes how to configure your bridge/router to establish serial line connectivity through X.25. This chapter also describes how this wide area protocol works and gives guidelines for operating, managing, and troubleshooting it.

The wide area bridge/router supports bridging and routing of multiple protocols over X.25. The X25 Service allows your bridge/router to transmit and receive data over an X.25 private or public data network (PDN). (See Figure 373)

Figure 373 X.25 Wide Area Protocol Over Serial Lines

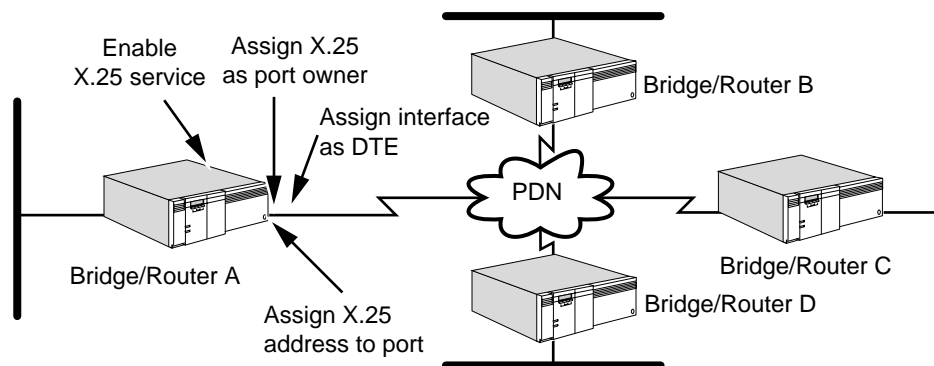


For conceptual information, see “How X.25 Works” later in this chapter.

Setting Up the X25 Service

This section describes how to configure your bridge/router to transmit and receive data over an X.25 interface. After you have completed these steps, proceed to “Setting Up Basic Routing over X.25” later in this chapter for routing configuration information.

Figure 374 and the procedures and examples that follow describe how to configure the X25 Service.

Figure 374 X.25 Configuration Overview

Prerequisites Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your wide area bridge/router ports and paths according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Obtain the X.25 address from the X.25 service provider. For more information, see “X25Address” in *Reference for Enterprise OS Software*.
- Determine if you have a meshed, partially meshed, or nonmeshed topology. If you have any of these topologies and plan to enable the Internet Protocol/Routing Information Protocol (IP-RIP), Internetwork Packet Exchange (IPX), or AppleTalk routing, ensure that the next-hop split horizon feature is enabled.

See “How X.25 Works” later in this chapter for information on meshed, partially meshed, and nonmeshed topologies; next-hop split horizon and how to configure it for IP-RIP, IPX, and AppleTalk; and virtual ports.

- If you plan to enable bridging or IP-OSPF (Internet Protocol/Open Shortest Path First), Xerox Network Systems (XNS), VINES, or DECnet IV routing, make sure that you have created a virtual port for each remote network that is attached to an X.25 cloud.

For instructions on setting up virtual ports, see the Configuring Advanced Ports and Paths chapter.

Procedure To enable your bridge/router to transmit and receive data over an X.25 PDN, follow these steps:

- 1 Assign X.25 as the owner of the path mapped to the port for each bridge/router port to be used for the X.25 Service using:

```
SETDefault !<port> -Port Owner = X25
```

- 2 Assign an X.25 data terminal equipment (DTE) address for each bridge/router port to be used for the X25 Service using:

```
SETDefault !<port> -X25 X25Address = <address>
```



Specify this parameter for a nonvirtual port only; do not specify it for a virtual port.

X.25 addresses are provided by the PDN at the time of subscription.

- 3 Adjust other X.25 parameters to suit your installation.

The default values for the X.25 parameters adhere to the default values for the X.25 standard. However, depending on the requirements of your installation, you may need to adjust parameters, such as X25PacketSize and X25WindowSize in the PROFile Service, and X25PROFileid in the X25 Service. Additional information about these parameters is described in the PROFile Service Parameters chapter in *Reference for Enterprise OS Software*.

CCITT X.25 specifications recommend that the logical channels used for virtual calls be configured in the following order: one-way incoming, two-way, and one-way outgoing. In this version, TwowaySVCs are currently configured for 1 through 4095 and the others are configured to NONE. For more information on these parameters, see the X25 Service Parameters chapter in *Reference for Enterprise OS Software*.

The bridge/router is configured by default for communication over a private data network. If you subscribe to one of the public data networks, specify the name of the PDN using:

```
SETDefault !<port> -X25 PDNetworkType = <pdnetworkname>
```

If you select a specific PDN, you may need to configure parameters to match the required settings of the PDN. The information on the specific settings of the PDN should be provided to you at when you subscribe to the PDN.

Verifying the Configuration

To verify the X.25 configuration, enter:

```
SHow -X25 CONFIguration
```

The bridge/router displays the current X.25 configuration information.

You can use the Trace parameter for debugging and troubleshooting purposes. For information on using Trace and CONFIguration parameters, see the X25 Service Parameters chapter in *Reference for Enterprise OS Software*.

Using X.25 Profiles

This section provides information about how X.25 profiles are used, and it briefly describes the X25 and PROFile Service parameters that help you configure source route bridging, transparent bridging, or routing over X.25.

The default characteristics for communicating over an X.25 interface is called the default DTE profile for a port. In general, the default characteristics provide optimum communications and no additional profiles are necessary. Under certain circumstances creating a profile may be advantageous. For example you may wish to create a profile under the following conditions:

- When the remote site has unique communication characteristics.
- When you have a closed user group.
- When you wish to use throughput class negotiation.

There are two types of profiles: X.25 DTE profiles and X.25 user profiles.

User Profiles

The X.25 user profile can be created and assigned to a specific port for a specific protocol. You use the X25PROFileid parameter to assign the profile to a specific port. If you are routing a network protocol such as IP or DECnet over X.25, you

can use the X.25 user profiles to qualify the type of virtual circuit over which the packet is forwarded.

When a user profile is assigned, all calls to and from the protocol on that port will use the X.25 parameters in that user profile. You must use the X.25 user profile parameters to reconfigure a virtual circuit for any network protocol. X.25 user parameters are a subset of X.25 DTE parameters and are listed in Table 91. The remaining X.25 parameters (not defined in the user profile) are taken from the default DTE profile for establishing a call, that is, the X.25 parameters in the user profile will overwrite the parameters in DTE profile for that call.

For example, AppleTalk wants to use a VCLimit of 4, IPX wants to use a VCLimit of 6, and other protocols want to use the default VCLimit in the DTE profile. A user profile can be created with an X25VCLimit set to 4 and another user profile can be created with an X25VCLimit set to 6. These profiles are then assigned to their respective protocols.

DTE Profiles The X.25 DTE profile contains a set of parameters that are used to establish a connection to a DTE. These parameters are listed in Table 91.

An X.25 DTE profile can be assigned to a specific port using the X25PROFileid parameter. The default DTE profile is assigned profile ID zero (0). All calls to and from the DTEs on a port use the DTE profile zero if the X25PROFileid parameter has not been configured for that port.

If you want to configure different X.25 parameters for different DTEs, you can create separate X.25 DTE profiles and assign each profile to a DTE using the -X25 NbrPROFile parameter. All the DTEs to which an X.25 DTE profile has not been assigned will use the default DTE profile.

For example, by default, incoming calls are allowed from all the DTEs. For security reasons, some DTEs may be allowed to establish the outgoing calls only. For those DTEs, you can create an X.25 DTE profile with NoIncomingCall and assign it to them.

X.25 Profile Parameter Usage

When an incoming call request is received, the incoming call facility parameters initially are compared with the configured DTE profiles. Before the call is accepted, a match must be found with a configured DTE profile or with the default DTE profile. Once the DTE profile is found, X.25 compares a subset of the incoming call facility parameters with the configured user profiles. The call facility parameters are X25PacketSiZe, X25ThruputClass, and X25WindowSiZe. Once the user profile is found, the user profile parameters (X25VCLimit, X25VCQueueSize, and X25VCTimer) are used to handle the congestion control for the virtual circuit. If a user profile is not found, the values from the matched DTE profile or the default DTE profile are used for congestion control.

Table 91 lists the X.25 DTE and X.25 user profile parameters.

Table 91 X.25 User and X.25 DTE Parameters

| X.25 User Profile Parameters | X.25 User Profile Parameter Default | X.25 DTE Profile Parameters | X.25 DTE Profile Parameters Default |
|------------------------------|-------------------------------------|-----------------------------|---|
| X25COMPResType | Default | X25COMPResType | Default |
| X25PacketSiZe | 128 | X25PacketSiZe | 128 |
| X25ProfileName | No default | X25ProfileName | No default |
| X25ThruputClass | 9600 | X25ThruputClass | 9600 |
| X25VCLimit | 2 | X25VCLimit | 2 |
| X25VCQueueSize | 10 | X25VCQueueSize | 10 |
| X25VCTimer | 5 | X25VCTimer | 5 |
| X25WindowSiZe | 2 | X25WindowSiZe | 2 |
| X25CUDSuffix | No default | X25ClosedUsrGrp | 0 |
| | | X25CONTRol | IncomingCall, OutgoingCall, NoPSN, NoWSN, NoTCN |
| | | X25FastSelect | NoRequest, NoAccess |
| | | X25ReverseCharge | NoRequest, Accept |

Configuration Parameters

This section describes the most useful parameters in configuring the characteristics of a particular DTE. Table 92 lists the parameters and services that can help you configure your bridge or bridge/router for the X.25 Service. For more complete information on these parameters, see the PROFile Service Parameters chapter and the X.25 Service Parameters chapter in *Reference for Enterprise OS Software*.

Table 92 X.25 Configuration Parameters

| Parameter | Service | Description |
|-------------------|---------|---|
| ProfileType | PROFile | Creates an X.25 profile that is used when X.25 virtual circuits are set up to carry bridge/router traffic. |
| X25Address | X25 | The international data number (IDN) assigned by the network provider. Can be up to 15 decimal digits. |
| X25PacketSiZe | PROFile | Specifies the packet size (in bytes) for a specified virtual circuit. |
| X25VCLimit | PROFile | Specifies the maximum number of virtual circuits to a specific DTE for a specific protocol. |
| X25VCQueueSize | PROFile | Specifies the maximum number of packets that can be queued for any single virtual circuit to a specific DTE when the virtual circuit on the X.25 port is congested. |
| X25VCThruputClass | PROFile | Specifies the throughput rate in bits per second. This parameter is used by the PDN to guarantee the bandwidth for the virtual circuit. |

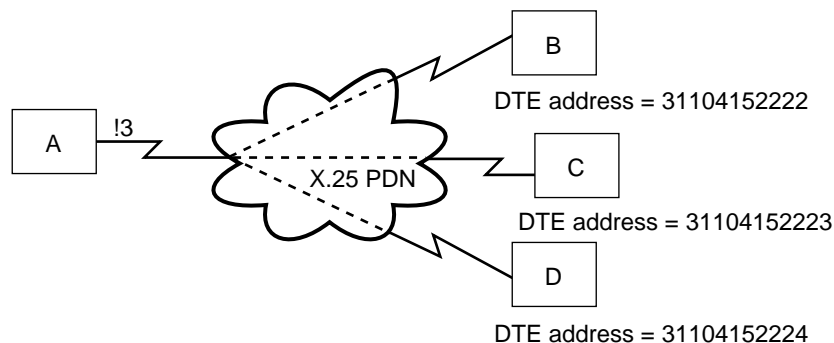
Table 92 X.25 Configuration Parameters

| Parameter | Service | Description |
|---------------|---------|---|
| X25VCTimer | PROFile | Specifies the maximum amount of time (in minutes) that can elapse when there is no activity on the X.25 virtual circuit before it is cleared. |
| X25WindowSiZe | PROFile | Determines the X.25 packet layer window size for the virtual circuit. |

X.25 Profiles Configuration Examples

This section provides examples of how X.25 DTE and X.25 user profiles can be applied to an X.25 network.

Example 1 Using Figure 375 as a sample network, assume you want to change the packet size and window size for all calls on port 3. You need to create an X.25 DTE profile with new packet size and window size values and assign them to port 3.

Figure 375 Creating X.25 DTE Profiles

To change the packet size and window size for all calls on port 3, follow these steps:

- 1 Create an X.25 DTE profile 4 by entering:
ADD !4 -PROFile ProfileType X25Dte
- 2 Increase the packet size in profile 4 from 128 (the default) to 1024 by entering:
SETDefault !4 -PROFile PacketSiZe = 1024
- 3 Increase the window size in profile 4 from 2 (the default) to 4 by entering:
SETDefault !4 -PROFile X25WindowSiZe = 4
- 4 Assign profile 4 to port 3 by entering:
SETDefault !3 -X25 X25PROFileid = 4

Example 2 In Figure 375, A is using port 3 to route IP and IPX to B, C, and D. You want to increase the throughput from A to B, from A and C, and from A to D. For security reasons, you want to allow B to establish only outgoing calls. To accomplish this, follow these steps:

- 1 Create an X.25 DTE profile 10 for A to B traffic by entering:
ADD !10 -PROFile ProfileType X25Dte
- 2 Create an X.25 DTE profile 20 for A to C traffic by entering:
ADD !20 -PROFile ProfileType X25Dte
- 3 Create an X.25 DTE profile 30 for A to D traffic by entering:

```
ADD !30 -PROfile ProfileType X25Dte
```

- Increase the throughput rate in profile 10 from 9600 (the default) to 19200 by entering:

```
SETDefault !10 -PROfile X25ThruputClass = 19200
```

- Allow B to establish only outgoing calls by entering:

```
SETDefault !10 -PROfile X25CONTRol = NoIncomingCall
```

- Increase the throughput rate in profile 20 from 9600 to 38400 by entering:

```
SETDefault !20 -PROfile X25ThruputClass = 38400
```

- Increase the throughput rate in profile 30 from 9600 to 48000 by entering:

```
SETDefault !30 -PROfile X25ThruputClass = 48000
```

- Assign profiles 10, 20, and 30 to B, C, and D, respectively, by entering:

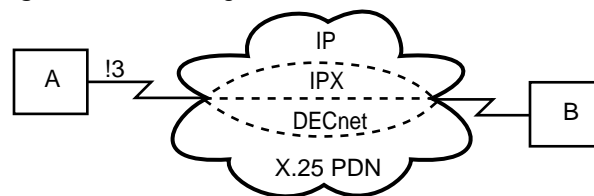
```
ADD -X25 NbrPROfile #31104152222 10
```

```
ADD -X25 NbrPROfile #31104152223 20
```

```
ADD -X25 NbrPROfile #31104152224 30
```

Example 3 You are routing IP, IPX, and DECnet between A and B over an X.25 PDN. You have been assigned six circuits for all traffic and want to allocate three to IP traffic, two to IPX, and one to DECnet. To allocate the traffic, see Figure 376 and follow these steps:

Figure 376 Creating X.25 User Profiles



- Create an X.25 user profile 15 to be used for routing IP traffic between A and B by entering:

```
ADD !15 -PROfile ProfileType X25User
```

- Create an X.25 user profile 25 to be used when routing IPX traffic between A and B by entering:

```
ADD !25 -PROfile ProfileType X25User
```

- Create an X.25 user profile 35 to be used when routing DECnet traffic between A and B by entering:

```
ADD !35 -PROfile ProfileType X25User
```

- Change the number of virtual circuits available for IP, IPX, and DECnet using the profiles established in steps 1–3 and by entering:

```
SETDefault !15 -PROfile X25VCLimit = 3
```

```
SETDefault !25 -PROfile X25VCLimit = 2
```

```
SETDefault !35 -PROfile X25VCLimit = 1
```

- Assign your X.25 user profiles to port 3 by entering:

```
SETDefault !3 -IP X25PROfileid = 15
```

```
SETDefault !3 -IPX X25PROfileid = 25
```

```
SETDefault !3 -DECnet X25PROfileid = 35
```

Example 4 Data prioritizing over X.25 does not use the Data Prioritizing scheme, that is the four levels of priority, as used by other WAN Services. This is because X.25 does not use LMF queuing. X.25 uses its own virtual circuit queue and maintains its own queues for each virtual circuit. If an IP data packet had a priority set to High under its global parameter, the bit would be set, but the X.25 queue would not check for this bit and would not place any priority on this packet.

Instead, X.25 uses X.25 user profiles to obtain the best bandwidth characteristics, that is the number of virtual circuits, the packet size, window size and so on. X.25 maintains its own queue and each virtual circuit can have different depths of queues and multiple queues per protocol. X.25 also does its own sequencing of packets and its own fragmentation. There are my X.25 parameters that are available to determine the number of virtual circuits per protocol, the queue size for each protocol, and the length of time the switched virtual circuit will stay open when there is not data. Other X.25 parameters give the ability to set the packet size, window size the throughput of each switched virtual circuit. These parameters allow for better control of the X.25 traffic.

User profiles for the IP service can be configured, so all traffic such as Telnet and FTP use the same profile. However, it is also possible to establish a user profile per IP protocol, that is one for Telnet and one for FTP. In this situation all the virtual circuits assigned for Telnet could be give better X.25 characteristics such as window size and throughput, compared with the switched virtual circuits assigned to FTP. In addition, all the IP protocol traffic could be given better X.25 characteristics than other protocol traffic. It also means that non-I/O traffic also gets a fair allocation of virtual circuits.

By default the IP protocol does not have an X.25 user profile configured. You much create an X.25 user profile if you want to assign a priority to IP packets over other traffic.

To prioritize FTP IP packets using and X.25 user profiles, follow these steps:

- 1 Create an X.25 user profile by entering:

```
Add !1 -PROfile ProfileType X25user
```

- 2 Assign the X.25 user profile the IP service by entering:

```
SETDefault !2 -IP X25profileid = 1
```

Giving the x25 profile an identity of 1, is an arbitrary number assigned by the user.

- 3 Adjust the number of virtual circuits for each profile ID by entering:

```
SETDefault !1 -PROfile X25VCLimit = 4
```

This allows IP protocols to use 4 virtual circuits, the default is 2.

- 4 Improve the response for FTP traffic by adjusting the X.25 window size by entering:

```
SETDefault !1 -PROfile X25WindowSize = 7
```

- 5 Further improve response for FTP traffic by adjusting the X.25 packet size by entering:

```
SETDefault !1 -PROfile X25PacketSize = 1024
```



When setting the X25PaketSIZE for Telnet traffic, be aware that Telnet will only use 64K packets so changing the size to larger than 64K will not help performance.

Setting Up Basic Routing over X.25

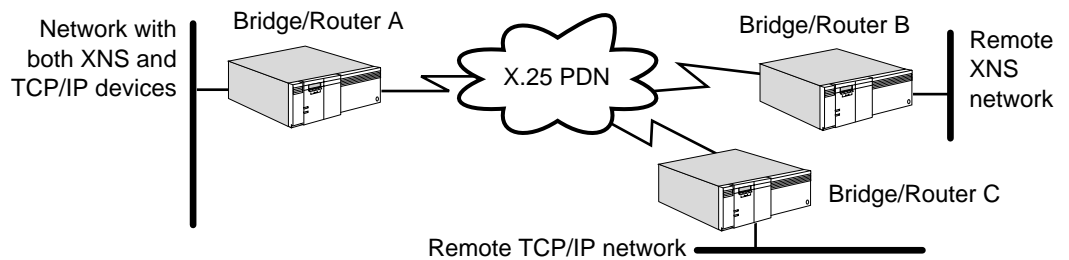
This section describes how to configure your router to transmit and receive data over an X.25 interface. Procedures for the following routing protocols are provided:

- AppleTalk
- Open System Interconnection (OSI)
- DECnet
- VINES
- IP
- Xerox Network Systems (XNS)
- IPX

A router can be configured to simultaneously route multiple protocols over X.25. For example, in Figure 377, the local network supports both XNS and TCP/IP traffic and routes information through a single X.25 connection to both types of remote networks.

If you are using X.25 to communicate with multiple routers over a single high-speed serial interface, you must have a fully meshed topology. Configure neighbors so the router can use next-hop split horizon to multiple routers on the same network, or use virtual ports where applicable.

Figure 377 Routing Multiple Protocols over X.25 PDN

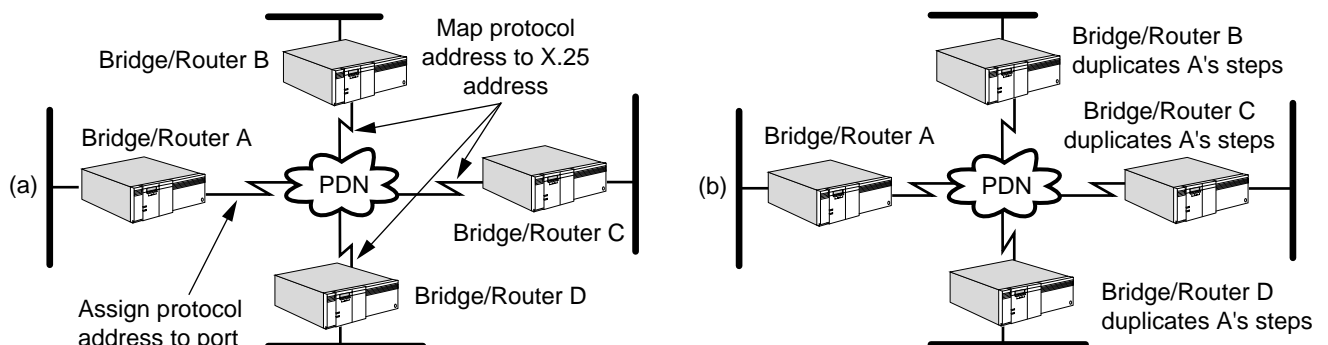


In this example, bridge/router A must be configured for operation with both XNS and TCP/IP, and the X.25 ports on the remote routers must be configured for their respective protocols.



Be sure each router attached to the PDN is configured with the same protocol ID.

Figure 378 Configuration Overview for Routing over X.25



Configuring AppleTalk

To allow the AppleTalk Protocol to operate over an X.25 PDN, you can configure the PDN to operate as either an AppleTalk or non-AppleTalk network. In both cases, the Routing Table Maintenance Protocol (RTMP) packet broadcasts are sent as directed broadcasts every 10 seconds (this is the default) to reach a router configured on a port.

The following section provides information for configuring AppleTalk routing for communication over an X.25 network.

For X.25 ports, split horizon decisions are made at the next router link level instead of at the port level. The next-hop split horizon feature allows support for nonmeshed topologies by allowing a router to use an X.25 port as a virtual hub, sending route information to each router out of the port learned from all other routers out of the same port. If the decisions were made at the port level, as for AppleTalk on LANs and SMDS, no routing information learned from any router out of the port will be sent to any router out of the same port.

Non-AppleTalk Prerequisites

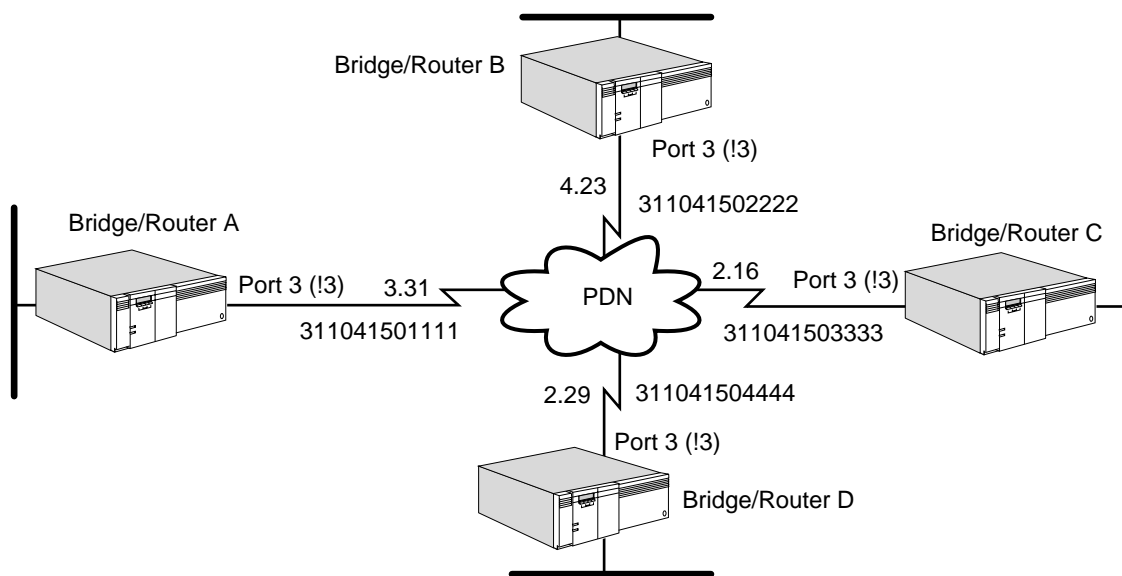
Before beginning this procedure, complete the following tasks:

- Configure your AppleTalk LAN according to the procedures in the Configuring AppleTalk Routing chapter.
- Set up the X25 Service as described in "Setting Up the X25 Service" earlier in this chapter.
- Obtain the X.25 addresses of each bridge/router participating in AppleTalk routing.

Non-AppleTalk Procedure

To configure AppleTalk routing over an X.25 PDN configured as a non-AppleTalk network, see Figure 379 and follow these steps:

Figure 379 Configuring AppleTalk over X.25



- 1 Configure all the ports on bridge/routers connected to the PDN to be connected to a non-AppleTalk network.

For example, on bridge/routers A, B, C, and D enter:

```
SETDefault !3 -AppleTalk CONTROL = NonAppleTalk
```

- 2 On each bridge/router, assign the X.25 address of the other bridge/routers connected to the PDN.

For example, on bridge/router A enter:

```
ADD -AppleTalk ADDRESS !3 #311041502222
ADD -AppleTalk ADDRESS !3 #311041503333
ADD -AppleTalk ADDRESS !3 #311041504444
```

Enter similar commands on bridge/routers B, C, and D.

You can dynamically add and delete VCs using the ADDRESS parameter.

- 3 Prioritize AppleTalk traffic over other protocols.

By default, the AppleTalk Protocol does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to AppleTalk packets over other traffic. To prioritize AppleTalk packets, follow these steps:

- a Use the -PROFILE ProfileType parameter to create an X.25 user profile.

See "ProfileType" in *Reference for Enterprise OS Software* and "X.25 Profiles Configuration Examples" earlier in this chapter for more information.

- b Assign the X.25 user profile to the AppleTalk Service using the X25PROFILEid parameter.

For example, suppose you want to use user profile 1 for carrying AppleTalk traffic over port 3. Enter:

```
SETDefault !3 -AppleTalk X25PROFILEid = 1
```

- 4 Specify a protocol identifier to be included in an outgoing X.25 call request.

By default, all NETBuilder bridge/routers use the hexadecimal value of 0xCA as the AppleTalk protocol identifier. This value ensures acceptance of an incoming call request when AppleTalk routing is enabled.

If you have a bridge/router from another vendor that needs to receive AppleTalk-routed packets, make sure that the protocol ID for all devices matches. You can change the value on the NETBuilder bridge/routers by using the X25ProtID parameter. For example, to change the value on port 3 of bridge/router A, enter:

```
SETDefault !3 -AppleTalk X25ProtID = 22
```

You can enter a hexadecimal value between 0 and FF.

- 5 Enable routing on each bridge/router by entering:

```
SETDefault !3 -AppleTalk CONTROL = ROute
```

AppleTalk Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your AppleTalk LAN according to the procedures in the Configuring AppleTalk Routing chapter.
- Set up the X25 Service as described in "Setting Up the X25 Service" earlier in this chapter.

- Obtain the AppleTalk node address and the X.25 address for each bridge/router participating in AppleTalk routing.

AppleTalk Procedure

To configure the X.25 PDN to operate as an AppleTalk network, see Figure 379 and follow this procedure.

Use the AppleTalk StartupNET and the StartupNODE commands to configure the local X.25 port's AppleTalk address. This allows the local router to always assign the same AppleTalk node address to the local port, assuming that the address is within the AppleTalk network range of the X.25 cloud. These static configurations are saved on the diskette and only need to be changed when the topology changes.

Set up mapping information between AppleTalk node addresses and X.25 addresses for each bridge/router directly connected to the PDN using the ADD -AppleTalk ADDRESS command.

The following sequence of commands sets up an AppleTalk network for an X.25 cloud with four routers (A–D) attached. This example assumes that the AppleTalk network range for the X.25 cloud shared by the configured routers is 2 to 4 and that at least one of the routers is configured to send seed information to any other nonseed routers.

To set up an AppleTalk network for an X.25 cloud, follow these steps:

- 1 Set the local AppleTalk address before routing is enabled by entering the following commands on bridge/router A:

```
SETDefault !3 -AppleTalk StartupNET = 3
SETDefault !3 -AppleTalk StartupNODE = 31
```

- 2 Set the local AppleTalk address before routing is enabled by entering the following commands on bridge/router B:

```
SETDefault !3 -AppleTalk StartupNET = 4
SETDefault !3 -AppleTalk StartupNODE = 23
```

- 3 Set the local AppleTalk address before routing is enabled by entering the following commands on bridge/router C:

```
SETDefault !3 -AppleTalk StartupNET = 2
SETDefault !3 -AppleTalk StartupNODE = 16
```

- 4 Set the local AppleTalk address before routing is enabled by entering the following commands on bridge/router D:

```
SETDefault !3 -AppleTalk StartupNET = 2
SETDefault !3 -AppleTalk StartupNODE = 29
```

- 5 Configure static mapping of neighbor X.25 DTE addresses to their AppleTalk node addresses on each bridge/router.

For example, on bridge/router A (AppleTalk address 3.31), enter the following X.25 addresses of the other bridge/routers connected to the PDN:

```
ADD -AppleTalk ADDRESS 4.23 #311041502222
ADD -AppleTalk ADDRESS 2.16 #311041503333
ADD -AppleTalk ADDRESS 2.29 #311041504444
```

Configure static mapping of media addresses on bridge/routers B (AppleTalk address 4.23), C (AppleTalk address 2.16), and D (AppleTalk address 2.29).

You can dynamically add and delete VCs using the ADDRess parameter.

- 6 Enable the X.25 ports on each router for routing over an AppleTalk network by using:

```
SETDefault !3 -AppleTalk CONTROL = (Route, AppleTalk)
```

Configuring DECnet

This section provides information for configuring DECnet routing for communication over an X.25 network.

Prerequisites

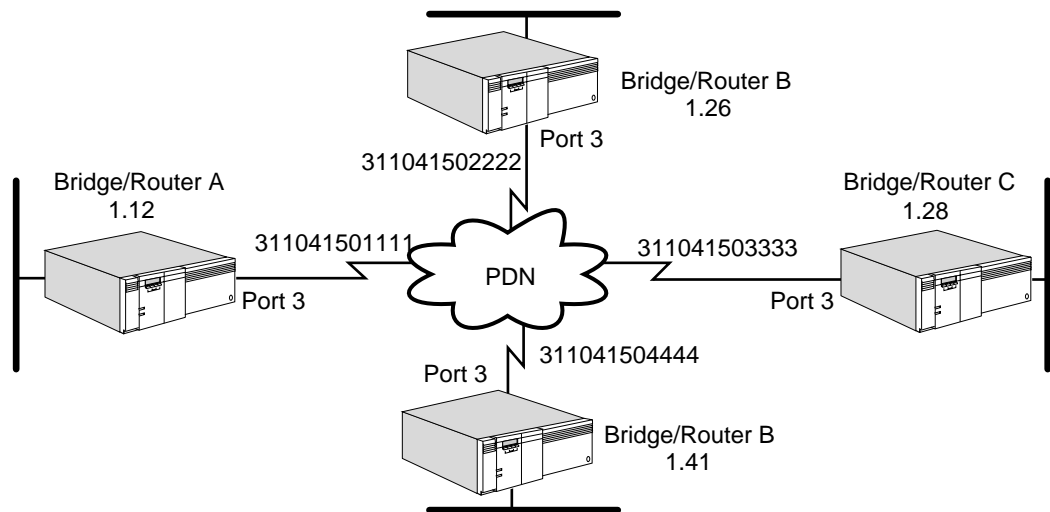
Before beginning this procedure, complete the following tasks:

- Configure your DECnet LAN according to the procedures in the Configuring DECnet Routing chapter.
- Set up the X25 Service as described in "Setting Up the X25 Service" earlier in this chapter.
- Obtain the DECnet address and X.25 address of each bridge/router participating in DECnet routing.

Procedure

To configure DECnet routing over an X.25 PDN, see Figure 380 and follow these steps.

Figure 380 Configuring DECnet over X.25



- 1 Set up mapping information between DECnet addresses and X.25 addresses for each bridge/router end node that is directly connected to the PDN.

Use the ADD !<port> -DECnet Neighbor syntax to set up mapping information. For example, on bridge/router A, enter:

```
ADD !3 -DECnet Neighbor 1.26 #311041502222
ADD !3 -DECnet Neighbor 1.28 #311041503333
ADD !3 -DECnet Neighbor 1.41 #311041504444
```

On bridge/routers B, C, and D, enter similar commands to specify the DECnet-to-X.25 address mapping information.



If you are configuring more than two neighbors, be sure that the X.25 parameters in the DECnet Service are configured as described in the remaining steps. For more information, see the DECnet Service Parameters chapter in Reference for Enterprise OS Software.

2 Prioritize DECnet traffic over other protocols.

By default, the DECnet Protocol does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to DECnet packets over other traffic. To prioritize DECnet packets, follow these steps:

- a Use the `-PROfile ProfileType` parameter to create an X.25 user profile.

See “ProfileType” in *Reference for Enterprise OS Software* and “X.25 Profiles Configuration Examples” earlier in this chapter for more information.

- b Assign the X.25 user profile to the DECnet Service using the `X25PROfileid` parameter.

For example, suppose you want to use user profile 1 for carrying DECnet traffic over port 3. Enter:

```
SETDefault !3 -DECnet X25PROfileid = 1
```

3 Specify a protocol identifier to be included in an outgoing X.25 call request.

By default, all NETBuilder II bridge/routers use the hexadecimal value of 0xDE as the DECnet protocol identifier. This value ensures acceptance of an incoming call request when DECnet routing is enabled.

If you have a bridge/router from another vendor that needs to receive DECnet routed packets, make sure that the protocol ID for all devices matches. You can change the value on the NETBuilder II bridge/routers by using the `X25ProtID` parameter. For example, to change the value on port 3 of bridge/router A, enter:

```
SETDefault !3 -DECnet X25ProtID = 33
```

You can enter a hexadecimal value between 0 and FF.

4 Enable DECnet routing on each port of each bridge/router that is attached to the X.25 PDN.

For example, to enable routing on port 3 of bridge/router A, enter:

```
SETDefault !3 -DECnet CONTROL = ROute
```

Enable routing on bridge/routers B, C, and D.

Configuring IP This section provides information for configuring IP routing over an X.25 network.

Prerequisites

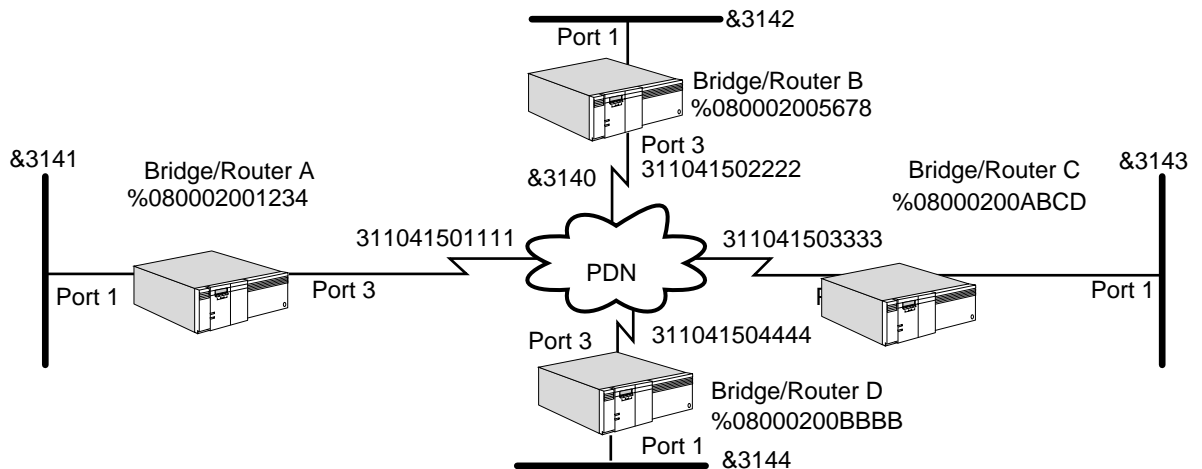
Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in the Configuring IP Routing chapter.
- Set up the X25 Service as described in “Setting Up the X25 Service” earlier in this chapter.
- Determine the IP addresses for each wide area port of your bridge/router that is attached to the X.25 network.
- Obtain the IP address and X.25 address of each bridge/router that is attached to the X.25 network and participating in IP routing.

Procedure

To enable IP over an X.25 network, see Figure 381 and follow these steps:

Figure 381 Configuring IP over X.25



- 1 Assign an IP address to each port on each 3Com router that is directly attached to the PDN.

The following command assigns the address 10.0.0.1 to port 3 on bridge/router A:

```
SETDefault !3 -IP NETaddr = 10.0.0.1
```

- 2 Specify IP to X.25 address mapping information for all neighbors.

The following sequence of commands specifies IP to X.25 address mapping information for the bridge/routers directly attached to the PDN. (In the examples that follow, DTE can be used in place of the pound sign [#].)

For example, enter the following commands on bridge/router A (IP address 10.0.0.1):

```
ADD -IP ADDRESS 10.0.0.2 #311041502222
ADD -IP ADDRESS 10.0.0.3 #311041503333
ADD -IP ADDRESS 10.0.0.4 #311041504444
```

Enter similar commands on bridge/router B (IP address 10.0.0.2), bridge/router C (IP address 10.0.0.3), and bridge/router D (IP address 10.0.0.4), specifying the IP address and DTE mapping information.

- 3 Optionally, if you are going to be running Open Shortest Path First (OSPF) as the routing protocol over X.25 switched virtual circuits, you can configure a demand interface circuit using:

```
SETDefault !<port> -OSPF DemandInterface = Enable
```



CAUTION: Do not configure any interface on any router in a single OSPF area as a demand circuit (DC) interface unless all routers in that area have been upgraded to at least software version 8.3.

With this setting, the router negotiates with the neighbor at the other end of the link. If the neighbor agrees that the link is a demand circuit, the router suppresses sending OSPF Hello packets and routing refresh information, allowing the data link connection to be closed when not carrying application traffic. In order for the demand circuit to be cost-effective, make sure that it is isolated from as many topology changes as possible because topology changes bring up the interface.

For more information, see “Reducing Network Costs Using Demand Interface Circuits” in the Configuring IP Routing chapter.

4 Enable the dynamic routing protocols using Routing Information Protocol-Internet Protocol (RIP-IP) or OSPF for each port and/or virtual port.

- To learn routes dynamically on port 3 using RIP, determine if the X.25 network is fully meshed or nonmeshed. If it is fully meshed, then enter:

```
SETDefault !3 -RIPIP CONTROL = (Talk, Listen, FullMesh)
```

If it is partially meshed or nonmeshed, enter the following command:

```
SETDefault !3 -RIPIP CONTROL = (Talk, Listen, NonMesh)
```

Setting the CONTROL parameter to the TALK and Listen values enables the router to send and receive routing information with other routers using RIP. If the FullMesh value is selected, RIP uses normal split horizon; if NonMesh is selected, RIP uses next-hop split horizon.



If the port owner is X.25, the port is up, and the -RIPIP CONTROL parameter is set to TALK, the DynamicNbr option for the -RIPIP and -OSPF CONTROL parameter are automatically enabled, which means that the software automatically adds neighbors and you can skip step 5 and proceed to step 6. If the NoDynamicNbr option for the CONTROL parameter is set, you must add neighbors by completing step 5.

- To enable routes dynamically on port 3 using OSPF, determine whether the X.25 network is fully meshed or nonmeshed.

If the network is fully meshed, enter:

```
SETDefault !3 -OSPF CONTROL = (Enable, FullMesh)
```

If the network is nonmeshed, enter:

```
SETDefault !3 -OSPF CONTROL = (Enable, NonMesh)
```

All OSPF neighboring routers must be configured with the same mode: FullMesh or NonMesh. Both modes apply to ports as well as virtual ports.

After OSPF operation is enabled, the router exchanges routing information with other routers using OSPF.

5 Specify neighbors for the routing protocols.

- a If your network is running RIP, add every router to which the configured router communicates to the neighbor list, either statically configured or learned dynamically.

For example, on bridge/router A, you must add the IP addresses of neighboring bridge/routers B, C, and D:

```
ADD !3 -RIPIP AdvToNeighbor 10.0.0.2  
ADD !3 -RIPIP AdvToNeighbor 10.0.0.3  
ADD !3 -RIPIP AdvToNeighbor 10.0.0.4
```

On bridge/router B, you must add the IP addresses of neighboring bridge/routers A, C, and D. In addition, add IP addresses of neighboring bridge/routers on bridge/routers C and D.

- b If your network is running OSPF, add every router to which the configured router communicates to the neighbor list, either statically configured or dynamically learned.

For example on bridge/router A, you must add the IP addresses of neighboring bridge/routers B, C, and D:

```
ADD !3 -OSPF Neighbor 10.0.0.2
```



```
ADD !3 -OSPF Neighbor 10.0.0.3
ADD !3 -OSPF Neighbor 10.0.0.4
```

On bridge/router B, you must add the IP addresses of neighboring bridge/routers A, C, and D. Also, add IP addresses of neighboring bridge/routers on bridge/routers C and D.

6 Prioritize IP traffic over other protocols.

By default, the IP Protocol does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to IP packets over other traffic. Currently, you can prioritize all IP packets or specific IP traffic based on IP filters. See “Filters” in *Reference for Enterprise OS Software* to create custom filters.

To prioritize IP packets using an X.25 user profile, follow these steps:

- a Use the `-Profile ProfileType` parameter to create an X.25 user profile. See “ProfileType” in *Reference for Enterprise OS Software* and “X.25 Profiles Configuration Examples” earlier in this chapter for more information.
- b Assign the X.25 user profile to the IP Service using the `X25PROFileid` parameter. For example, suppose you want to use user profile 1 for carrying IP traffic over port 3. Enter:

```
SETDefault !3 -IP X25PROFileid = 1
```

If a user profile is configured for the IP Service (an IP user profile ID), all IP traffic uses the IP user profile ID. You can also prioritize traffic using the `X25Profile` action in the `FilterAddr` parameter. For example, you can set the `FilterAddr` parameter to select different user profile IDs that prioritize Telnet traffic over FTP. The user profiles configured using the `FilterAddr` parameter overwrite the IP user profile ID. When separate user profiles are configured for Telnet/FTP traffic using filters, Telnet and FTP can establish separate virtual circuits to carry the traffic, guaranteeing that FTP packets will not take over the virtual circuits. You can adjust the `X25WindowSize` and `X25PacketSize` parameters in the user profile to improve the response of Telnet traffic over X.25.

7 Specify a protocol identifier to be included in an outgoing X.25 call request.

By default, all NETBuilder II bridge/routers use the hexadecimal value of 0xCC as the IP protocol identifier. This value ensures acceptance of an incoming call request when IP routing is enabled.

If you have a bridge/router from another vendor that needs to receive IP-routed packets, make sure that the protocol ID for all devices matches. You can change the value on the NETBuilder II bridge/routers by using the `X25ProtID` parameter. For example, to change the value on port 3 of bridge/router A, enter:

```
SETDefault !3 -IP X25ProtID = 44
```

You can enter a hexadecimal value between 0 and FF.

Configuring IPX

This section provides information for configuring IPX routing for communication over an X.25 network.

Prerequisites

Before beginning this procedure, complete the following tasks:

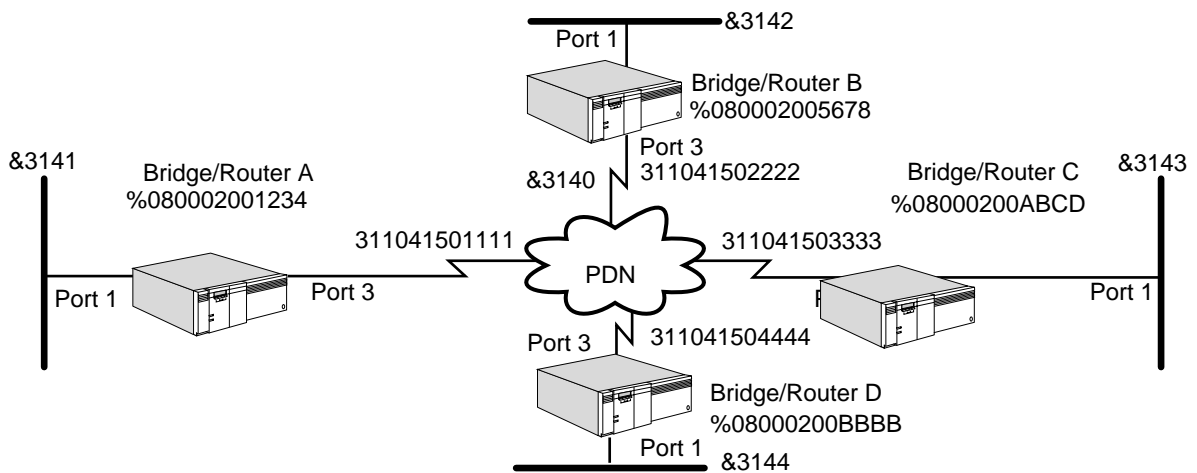
- Configure your IPX LAN according to the procedures in the Configuring IPX Routing chapter.

- Set up the X25 Service as described in “Setting Up the X25 Service” earlier in this chapter.
- Determine the IPX network number to be assigned to each bridge/router.

Procedure

To configure IPX over an X.25 PDN, see Figure 382 and follow these steps:

Figure 382 Configuring IPX over X.25



- 1 Assign a network number to each port on each 3Com bridge/router connected to the X.25 PDN.

For example, assign &3140 as the network number to port 3 on bridge/routers A, B, C, and D by entering the following command on each bridge/router:

```
SETDefault !3 -IPX NETnumber = &3140
```

- 2 Specify IPX network number to X.25 address mapping information for each bridge/router directly connected to the PDN.

For example, on bridge/router A, enter:

```
ADD !3 -IPX ADDRESS #311041502222 %080002005678
ADD !3 -IPX ADDRESS #311041503333 %08000200ABCD
ADD !3 -IPX ADDRESS #311041504444 %08000200BBBB
```

The commands specify IPX to X.25 address mapping information; the network number in each case corresponds to port 3 on the remote bridge/router.

Enter similar commands on bridge/routers B, C, and D.

- 3 If you are using NetWare Routing Information Protocol (NRIP) and Service Advertising Protocol (SAP) as your routing protocols, verify that routing is enabled on each wide area port of each bridge/router that is attached to the X.25 network by entering:

```
SHow -NRIP CONTrol
```

To verify that Talk and Listen are set, enter the SHow -SAP CONTROL command.

- 4 If you are using NetWare Link Services Protocol (NLSP) as the routing protocol, follow these steps:

- a Make sure the NLSP routing protocol is enabled by entering:

```
SHow -NLSP CONTrol
```

- b** Skip this step if dynamic neighbor is enabled on the port. Specify the DTE address neighbors that will be taking part in routing over X.25 using:

```
ADD !<port> -NLSP Neighbors #<DTE address>
```

For example on bridge/router A, enter the DTE address of bridge/routers B, C, and D as follows:

```
ADD !3 -NLSP Neighbors #311041502222
ADD !3 -NLSP Neighbors #311041503333
ADD !3 -NLSP Neighbors #311041504444
```

- c** Display the NLSP adjacencies by entering:

```
SHow -NLSP ADJAcencies
```



If you are configuring more than two neighbors, be sure that the X.25 parameters in the PROFile Service are configured as described in the remaining steps. For more information, see the PROFile Service Parameters chapter in Reference for Enterprise OS Software.

- 5** Prioritize IPX traffic over other protocols.

By default, the IPX Protocol does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to IPX packets over other traffic. To prioritize IPX packets, follow these steps:

- a** Use the -PROFile ProfileType parameter to create an X.25 user profile.

See "ProfileType" in *Reference for Enterprise OS Software* and "X.25 Profiles Configuration Examples" earlier in this chapter for more information.

- b** Assign the X.25 user profile to the IPX Service using the X25PROFileid parameter.

For example, suppose you want to use user profile 1 for carrying IPX traffic over port 3. Enter:

```
SETDefault !3 -IPX X25PROFileid = 1
```

- 6** Specify a protocol identifier to be included in an outgoing X.25 call request.

By default, all 3Com bridge/routers use the hexadecimal value of 0xEE as the IPX protocol identifier. This value ensures acceptance of an incoming call request from other 3Com routers.

If you have a bridge/router from another vendor that needs to receive IPX-routed packets, make sure that the protocol IDs are compatible. You can change the value on the 3Com bridge/routers by using the X25ProtID parameter. For example, to change the value on port 3 of bridge/router A, enter:

```
SETDefault !3 -IPX X25ProtID = 55
```

You can enter a hexadecimal value between 0 and FF.

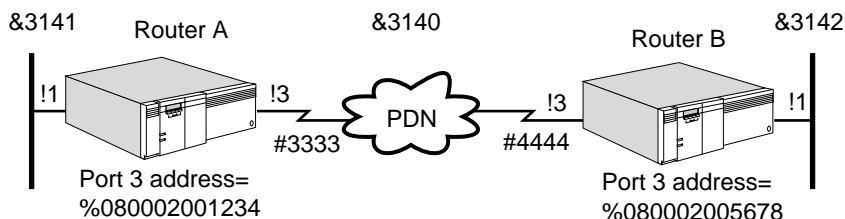
You can force the 3Com bridge/router to comply with the RFC 1356 by setting the value to Internet Engineering Task Force (IETF). For example, to change the router to IETF compliancy, enter:

```
SETDefault !3 -IPX X25ProtID = IETF
```

Configuring IPX with Different Software Versions

To configure IPX to operate over an X.25 PDN when bridge/router A is running 6.0 software or later and bridge/router B is running a version earlier than 6.0, see Figure 383 and follow these steps.

Figure 383 Configuring IPX with Different Software Versions



On bridge/router A, follow these steps:

- 1 Assign a network number to the port that is connected to the X.25 PDN.
For example, assign &3140 as the NETnumber to port 3 on bridge/router A by entering:

```
SETDefault !3 -IPX NETnumber = &3140
```

- 2 Configure bridge/router A to interoperate with software earlier than 6.0 by using the ripConTRoL parameter:

```
SETDefault !3 -IPX ripConTRoL = OldNbrMap
```

In software release 8.0 and later, use:

```
SETDefault !3 -NRIP CONTrol = OldNbrMap
```

- 3 Specify an IPX network number to X.25 address mapping information for the bridge/router A port that is directly connected to the PDN.

Using Figure 383 as an example, enter:

```
ADD !3 -IPX ADDRESS #4444 %080002005678
```

The address is optional.

On bridge/router B, follow these steps:

- 1 Assign a network number to the port that is connected to the X.25 PDN.
Using Figure 383 as an example, assign &3140 as the network number to port 3 on bridge/router B by entering:

```
SETDefault !3 -IPX NETnumber = &3140
```

- 2 Specify an IPX network number to X.25 address mapping information for the bridge/router B port that is directly connected to the PDN.

Use Figure 383 as an example, enter:

```
ADD !3 -IPX ADDRESS &3141 #3333
```

When adding a neighbor to bridge/router B, you must assign the Router A port 1 network number (&3141) to the bridge/router B port.

Configuring OSI This section provides information for configuring OSI routing for communication over an X.25 network.

Prerequisites

Before beginning the procedure, decide whether to use the PrefixRoute parameter or the Neighbors parameter using the following criteria:

- Use the PrefixRoute parameter if you view the remote site as another routing domain (for example, another company) with different NSAP addresses. The PrefixRoute parameter allows you to specify interdomain reachability information without exchanging Intermediate System-to-Intermediate System (IS-IS) packets.
- Use the Neighbors parameter if the remote site is part of your routing domain. The neighbor information instructs the IS-IS Protocol to exchange packets and establish full connectivity.

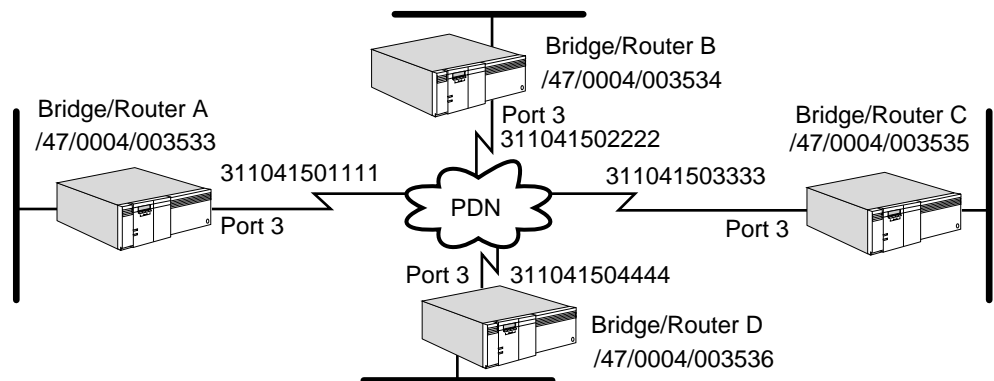
In addition, you need to complete the following tasks:

- Configure your OSI LAN according to the procedures in the Configuring Wide Area Networking Using X.25 chapter.
- Set up the X25 Service as described in “Setting Up the X25 Service” earlier in this chapter.
- If you are using the PrefixRoute parameter, obtain the NSAP address prefix and the X.25 address for each bridge/router participating in OSI routing.
- If you are using the Neighbors parameter, obtain the X.25 address of each bridge/router participating in OSI routing.

Procedure

To configure OSI routing, see Figure 384 and follow these steps. If you want to use the PrefixRoute parameter, begin with step 1. If you want to use the Neighbors parameter, skip step 1 and begin with step 2.

Figure 384 Configuring OSI over X.25



- Using the PrefixRoute parameter, specify an OSI address prefix and corresponding X.25 The MODE parameter in the ISIS Service must be set to L2 for the PrefixRoute parameter to take effect.

For example, on bridge/router A, enter:

```
ADD !3 -ISIS PrefixRoute /47/0004/003534 #311041502222
ADD !3 -ISIS PrefixRoute /47/0004/003535 #311041503333
ADD !3 -ISIS PrefixRoute /47/0004/003536 #311041504444
```

Enter similar commands on bridge/router B, C, and D, specifying OSI-to-X.25 address mapping information.

Proceed to step 3.

- Using the Neighbors parameter, specify an X.25 address for any neighbors on the X.25 PDN that support IS-IS.

IS-IS operates over X.25 in a point-to-point manner and does not require a fully meshed connectivity between all the bridge/routers.

Using Figure 384 as an example, if bridge/router B supports IS-IS and you want to operate it over X.25, you would enter the following command from bridge/routers A, C, and D:

```
ADD !3 -ISIS Neighbors #311041502222
```

On bridge/router B, enter:

```
ADD !3 -ISIS Neighbors #311041501111
ADD !3 -ISIS Neighbors #311041503333
ADD !3 -ISIS Neighbors #311041504444
```

- Prioritize OSI traffic over other protocols.

By default, the OSI Protocol does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to OSI packets over other traffic. To prioritize OSI packets, follow these steps:

- Use the -PROfile ProfileType parameter to create an X.25 user profile.
See "ProfileType" in *Reference for Enterprise OS Software* and "X.25 Profiles Configuration Examples" earlier in this chapter for more information.
- Assign the X.25 user profile to the OSI Service using the X25PROfileid parameter.

For example, suppose you want to use user profile 1 for carrying OSI traffic over port 3. Enter:

```
SETDefault !3 -CLNP X25PROfileid = 1
```

Configuring VINES

This section provides information for configuring VINES routing for communication over an X.25 network.

Prerequisites

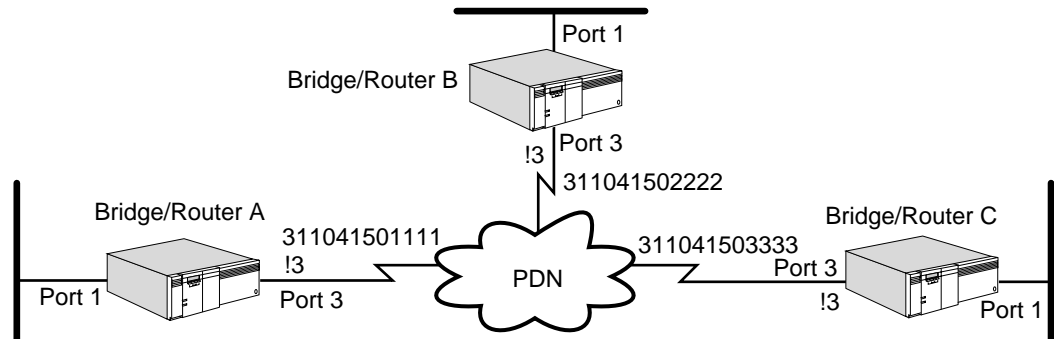
Before beginning this procedure, complete the following tasks:

- Configure your VINES LAN according to the procedures in the Configuring Wide Area Networking Using X.25 chapter.
- Set up the X25 Service as described in "Setting Up the X25 Service" earlier in this chapter.
- Obtain the X.25 addresses of each bridge/router participating in VINES routing.

Procedure

To enable the VINES Protocol to operate over an X.25 PDN, see Figure 385 and follow these steps:

Figure 385 Configuring VINES over X.25



- 1 Specify X.25 DTE addresses for port or virtual ports.

For example, on bridge/router A, enter:

```
ADD !3 -VIP WideAreaNbr #311041502222
ADD !3 -VIP WideAreaNbr #311041503333
```

Enter similar commands on bridge/routers B and C, specifying the DTE addresses for the ports.

- 2 Prioritize VINES traffic over other protocols.

By default, the VINES Protocol does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to VINES packets over other traffic. To prioritize VINES packets, follow these steps:

- a Use the -PROFILE ProfileType parameter to create an X.25 user profile.

See "ProfileType" in *Reference for Enterprise OS Software* and "X.25 Profiles Configuration Examples" earlier in this chapter for more information.

- b Assign the X.25 user profile to the VINES Service using the X25PROFILEid parameter.

For example, suppose you want to use user profile 1 for carrying VINES traffic over port 3. Enter:

```
SETDefault !3 -VIP X25PROFILEid = 1
```

- 3 Specify a protocol identifier to be included in an outgoing X.25 call request.

By default, all NETBuilder II bridge/routers use the hexadecimal value of 0xBC as the VINES protocol identifier. This value ensures acceptance of an incoming call request when VINES routing is enabled.

If you have a bridge/router from another vendor that needs to receive VINES-routed packets, make sure that the protocol ID for all devices matches. You can change the value on the NETBuilder II bridge/routers by using the X25ProtID parameter. For example, to change the value on port 3 of bridge/router A, enter:

```
SETDefault !3 -VIP X25ProtID = 66
```

Configuring XNS

The section provides information for configuring XNS routing for communication over an X.25 network.

Prerequisites

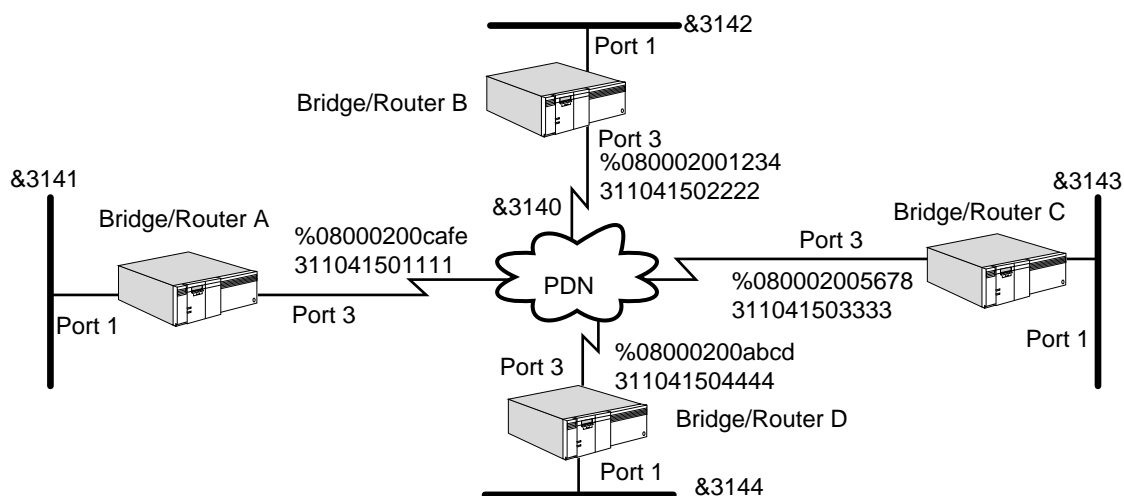
Before beginning this procedure, complete the following tasks:

- Configure your XNS LAN according to the procedures in the Configuring XNS Routing chapter.
- Set up the X25 Service as described in “Setting Up the X25 Service” earlier in this chapter.
- Determine the XNS network number to be assigned to each bridge/router.
- Obtain the MAC address and X.25 address of each remote host participating in XNS routing.

Procedure

To enable the XNS Protocol to operate over an X.25 PDN, see Figure 386 and follow these steps:

Figure 386 Configuring XNS over X.25



- 1 Assign a network number to each port on each 3Com router that is connected to the X.25 PDN.

For example, to assign &3140 as the network number to port 3 on bridge/routers A, B, C, and D, enter the following command on each router:

```
SETDefault !3 -IDP NETnumber = &3140
```

- 2 Set up mapping information between NETnumber and X.25 addresses for each bridge/router directly connected to the PDN.

Using Figure 386 as an example, the following sequence of commands specifies network number to X.25 address mapping information. The network number in each case corresponds to port 3 on the remote bridge/router.

For example, enter the following commands on bridge/router A:

```
ADD !3 -RIPXNS ADDRESS %080002001234 #311041502222
ADD !3 -RIPXNS ADDRESS %080002005678 #311041503333
ADD !3 -RIPXNS ADDRESS %08000200abcd #311041504444
```

Enter similar commands on bridge/routers B, C, and D, specifying the MAC address and the X.25 address mapping information.

3 Prioritize XNS traffic over other protocols.

By default, the XNS Protocol does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to XNS packets over other traffic. To prioritize XNS packets, follow these steps:

a Use the -Profile ProfileType parameter to create an X.25 user profile.

See "ProfileType" in *Reference for Enterprise OS Software* and "X.25 Profiles Configuration Examples" earlier in this chapter for more information.

b Assign the X.25 user profile to the IDP Service using the X25PROFileid parameter.

For example, suppose you want to use user profile 1 for carrying XNS traffic over port 3. Enter:

```
SETDefault !3 -IDP X25PROFileid = 1
```

4 Specify a protocol identifier to be included in an outgoing X.25 call request.

By default, all NETBuilder II bridge/routers use the hexadecimal value of 0xC0 as the XNS protocol identifier. This value ensures acceptance of an incoming call request when XNS routing is enabled.

If you have a bridge/router from another vendor that needs to receive XNS-routed packets, make sure that the protocol ID for all devices matches. You can change the value on the NETBuilder II bridge/routers by using the X25ProtID parameter. For example, to change the value on port 3 of bridge/router A, enter:

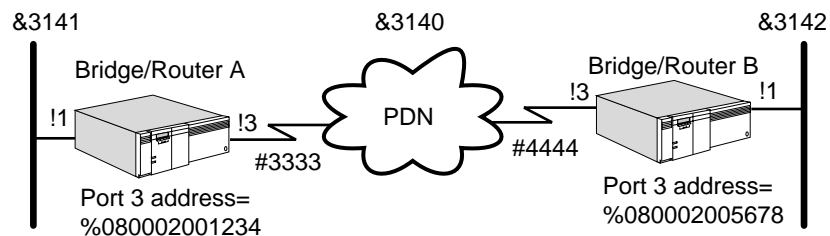
```
SETDefault !3 -IDP X25ProtID = 77
```

You can enter a hexadecimal value between 0 and FF.

Procedure

To configure XNS to operate over an X.25 PDN when bridge/router A is running 5.0 software or later and bridge/router B is running an earlier version, see Figure 387 and follow these steps:

Figure 387 Enabling XNS Across a PDN Between Two Neighbors With Different Software



Versions

On bridge/router A, follow these steps:

1 Assign a network number to the port that is connected to the X.25 PDN.

Assign &3140 as the network number to port 3 on bridge/router A by entering:

```
SETDefault !3 -IDP NETnumber = &3140
```

- 2 Configure bridge/router A to interoperate with software earlier than 5.0 by entering:

```
SETDefault !3 -RIPXNS CONTROL = OldNbrMap
```

- 3 Specify XNS-to-X.25 address mapping information for the bridge/router A port that is directly connected to the PDN by entering the following command:

```
ADD !3 -RIPXNS ADDRESS %080002005678 #4444
```

On bridge/router B, follow these steps:

- 1 Assign a network number to the port that is connected to the X.25 PDN.

For example, to assign &3140 as the network number to port 3 on bridge/router B, enter:

```
SETDefault !3 -IDP NETnumber = &3140
```

- 2 Specify XNS-to-X.25 address mapping information for the bridge/router B port that is directly connected to the PDN.

For example, use the following command to specify the XNS-to-X.25 address mapping information.

```
ADD !3 -RIPXNS ADDRESS &3141 #3333
```



When adding a neighbor on bridge/router B, it must use the network number of port 1 on bridge/router A.

The NETBuilder II bridge/router by default specifies addresses in canonical format, and a SuperStack II NETBuilder bridge/router model 327 or 527 by default specifies addresses in noncanonical format. When connecting the two platforms using an X.25 link running XNS, the NETBuilder II bridge/router will not know that the model 327 or 527 bridge/router is a token ring platform. The token ring models will not know that the NETBuilder II bridge/router is an Ethernet platform. You must configure each platform as a static neighbor to the other platform and specify the neighbor's address in canonical format for Ethernet and noncanonical format for token ring. Use:

```
ADD !<port> -RIPXNS ADDRESS %<host> <media address>
```

When using this syntax on the NETBuilder II bridge/router, you must specify the remote host address in noncanonical format to indicate that the remote host is a token ring platform (model 327). When using this syntax on a model 327 bridge/router, you must specify the remote host address in canonical format to indicate that the remote host is an Ethernet platform (NETBuilder II bridge/router).

Setting Up Bridging over X.25

This section describes how to configure your bridge to forward packets over X.25.

Bridging over X.25 requires two or more 3Com bridges to be connected over one or more X.25 PDNs to access nodes on remote LANs. The bridge will not learn from DTEs that are not preconfigured as a neighbor.

Configuring Transparent Bridging

This section describes how to configure transparent bridging.

Prerequisites

Before beginning this procedure, complete the following tasks:

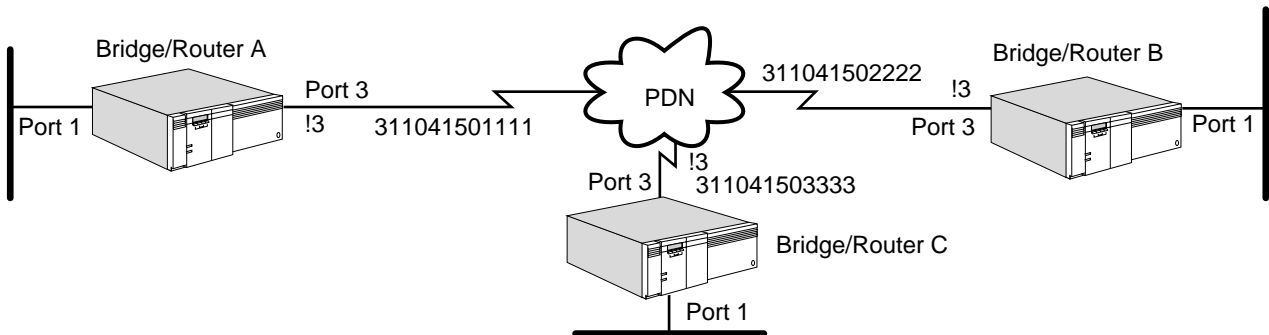
- Configure your LAN according to the transparent bridging procedures in the Configuring Bridging chapter.

- Set up the X25 Service as described in “Setting Up the X25 Service” earlier in this chapter.
- Obtain the X.25 addresses of each bridge/router participating in transparent bridging.

Procedure

To configure transparent bridging over an X.25 PDN, see Figure 388 and follow these steps:

Figure 388 Configuring Transparent Bridging over X.25



- 1 Verify that transparent bridging is enabled on each 3Com bridge port that is directly connected to the X.25 PDN.

By default, transparent bridging is enabled on all NETBuilder II bridge/routers. To verify the setting, use:

```
SHow [!<port>] -BRidge TransparentBRidge
```

If transparent bridging has been disabled, you can enable it on port 3 of Bridge/router A, B, and C by entering the following command on each of these devices:

```
SETDefault !3 -BRidge TransparentBRidge = TransparentBRidge
```

- 2 Enable the bridge by entering:

```
SETDefault -BRidge CONTROL = Bridge
```

- 3 Configure all DTEs on the PDN as neighbors that will take part in bridging over X.25.



Perform this step for nonvirtual ports only.

You can configure a maximum of eight neighbors per port.

To configure a neighbor, on bridge/router A enter:

```
ADD !3 -BRidge X25Neighbor = 311041502222
```

```
ADD !3 -BRidge X25Neighbor = 311041503333
```

Enter similar commands on bridge/routers B and C to configure the DTEs on the PDN as neighbors.

- 4 Prioritize bridge traffic.

By default, the BRidge Service does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to bridged packets over other traffic. To prioritize bridged packets, follow these steps:

- a Use the -PROfile ProfileType parameter to create an X.25 user profile.

See "ProfileType" in *Reference for Enterprise OS Software* and "X.25 Profiles Configuration Examples" earlier in this chapter for more information.

- b Assign the X.25 user profile to the BRidge Service using the X25PROFileid parameter.

For example, suppose you want to use user profile 1 for carrying the bridged traffic over port 3. Enter:

```
SETDefault !3 -BRidge X25PROFileid = 1
```

With the current X25VCLimit default, the BRidge Service can establish more than one virtual circuit to a destination. Because the number of virtual circuits is greater than one, packets may not be received in the order in which they were sent. For some bridge-only protocols, such as local area transport (LAT), the sequence of packets needs to be maintained. If the bridged environment consists of these types of protocols, you must create an X.25 user profile with the X25VCLimit parameter set to 1, and assign this profile ID in the BRidge Service. Mnemonic filters can be used to prioritize bridged traffic over X.25. For example, you can configure mnemonic filters for IP and IPX. You can also assign user profiles that are different from the bridge profile ID. All bridged IP and IPX traffic can establish separate virtual circuits for carrying the traffic. Remaining bridged traffic uses the bridge user profile ID.

- 5 Specify a protocol identifier to be included in an outgoing X.25 call request.

By default, all NETBuilder II bridge/routers use the hexadecimal value of 0xDD as the transparent bridging protocol identifier. This value ensures acceptance of an incoming call request when transparent bridging is enabled.

If you have a bridge/router from another vendor that needs to receive transparent bridging packets, make sure that the protocol ID for all devices matches. You can change the value on the NETBuilder II bridge/routers by using the X25ProtID parameter. For example, to change the value on port 3 of bridge/router A, enter:

```
SETDefault !3 -BRidge X25ProtID = 11
```

You can enter a hexadecimal value between 0 and FF.

Configuring Source Route Bridging

This section provides information for configuring source route bridging over X.25. For more information about source route bridging, see the Configuring Source Route Bridging chapter.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the source route bridging procedures in the Configuring Source Route Bridging chapter.
- Set up the X25 Service as described in "Setting Up the X25 Service" earlier in this chapter.
- Obtain the X.25 addresses of each bridge/router participating in source route bridging.

Procedure

To configure source route bridging over X.25, follow these steps:

- 1 Configure all DTEs on the source routing X.25 port as neighbors using this syntax:

```
ADD !<port> -BRidge X25Neighbor = <address>
```



You can configure a maximum of eight neighbors per port.

Perform this step for virtual ports only.

On bridge/router A, enter

```
ADD !3 -BRidge X25Neighbor = 311041502222
ADD !3 -BRidge X25Neighbor = 311041503333
```

Enter similar commands on bridge/routers B and C to configure the DTEs on the PDN as neighbors.

- 2 Assign a unique ring number to the logical ring associated with each X.25 source routing port.

The ring number can be any number in the range 1 to 4,095, and can be entered in either decimal or hexadecimal format using:

```
SETDefault !<port> -SR RingNumber = <number>(1-4095) |
0x<number>(1-FFF)
```

You can enter the ring number in decimal or hexadecimal format. Precede the hexadecimal number with 0x.

For more information about ring numbers, see the Configuring Wide Area Networking Using X.25 chapter in *Reference for Enterprise OS Software*.

- 3 Prioritize bridge traffic.

By default, the BRidge Service does not have an X.25 user profile configured. Configure X.25 user profiles only if you want to assign a priority to bridged packets over other traffic. To prioritize bridged packets, follow these steps:

- a Use the -PROfile ProfileType parameter to create an X.25 user profile.
See "ProfileType" in *Reference for Enterprise OS Software* and "X.25 Profiles Configuration Examples" earlier in this chapter for more information.
- b Assign the X.25 user profile to the BRidge Service using the X25PROFileid parameter.

For example, suppose you want to use user profile 1 for carrying the bridged traffic over port 3. Enter:

```
SETDefault !3 -BRidge X25PROFileid = 1
```

With the current X25VCLimit default, the BRidge Service can establish more than one virtual circuit to a destination. Because the number of virtual circuits is greater than one, packets may not be received in the order in which they were sent. For some bridge-only protocols, such as LAT, the sequence of packets needs to be maintained. If the bridged environment consists of these types of protocols, you must create an X.25 user profile with the X25VCLimit parameter set to 1, and assign this profile ID in the BRidge Service.

Mnemonic filters can be used to prioritize bridged traffic over X.25. For example, you can configure mnemonic filters for IP and IPX. You can also assign user profiles that are different from the bridge profile ID. All bridged IP and IPX traffic can establish separate virtual circuits for carrying the traffic. Remaining bridged traffic uses the bridge user profile ID.

- 4 Verify that source route bridging is enabled on the wide area port using:

```
SHow !<port> -SR SrcRouBridge
```

If source route bridging is disabled, you need to enable it for your wide area port:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```

- 5 If you want to run both source route and transparent bridging on a NETBuilder II bridge/router, skip this step and go on to step 6. If you want to run source route bridging only on a NETBuilder II bridge/router, disable transparent bridging on the wide area port using:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

This step does not apply to model 32x and 52x SuperStack II NETBuilder bridge/router. Transparent bridging is not supported on these models.

- 6 Verify that bridging is enabled by entering:

```
SHow -BRidge CONFIguration
```

If bridging has been disabled, enable it for the system by entering:

```
SETDefault -BRidge CONTrol = Bridge
```

Setting Up a Permanent Virtual Circuit Connection

This section describes how to set up permanent virtual circuits (PVC) on an X.25 interface. A fixed point-to-point connection can use a PVC to emulate a leased or private line. X.25 PVCs can be set up on routed configurations to transmit and receive data over an X.25 interface on public data networks.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the X25 Service as described in "Setting Up the X25 Service" earlier in this chapter.
- Obtain the X.25 addresses of the destination bridge/router participating in the PVC.
- Create an X.25 user profile to assign the target DTE for a the desired routing protocol.

Procedure

To configure an X.25 PVC, follow these steps:

- 1 Add the port using:

```
ADD! <port> -X25 PVC <lcnl> [,lcnl2] <destination dte address>
<protocol ID> [<user profID>]
```

For example, to set up a PVC on port 2 of your bridge/router, enter:

```
ADD !2 -X25 PVC 1,2 311022255731 CC
```

This command creates a PVC connection on port 2. This PVC carries IP traffic, specified by protocol ID CC, to and from DTE address 311022255731 on logical channel numbers 1 and 2.

- 2 To verify the X.25 PVC configuration, enter:

```
SHow !2 -X25 PVC
```

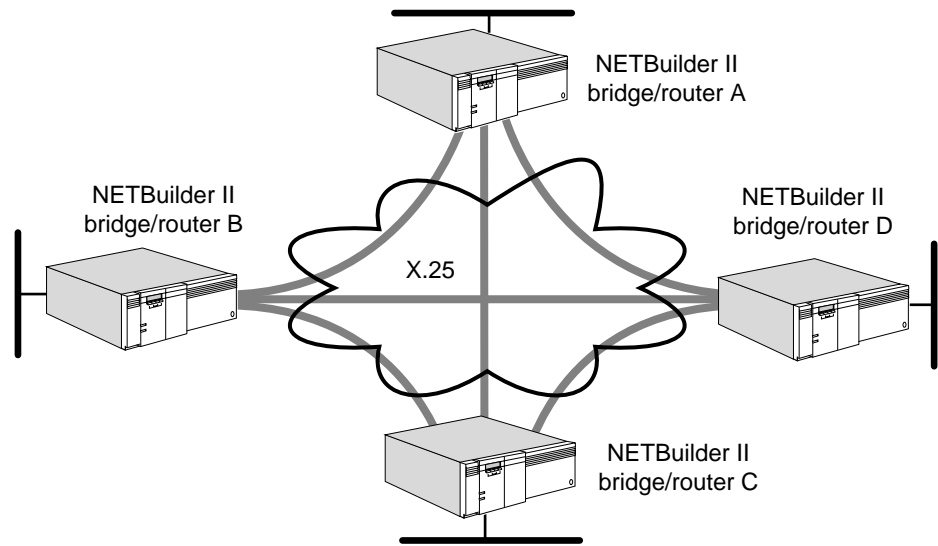
The PVCs configured on port 2 are displayed.

How X.25 Works

This section describes the X25 Service.

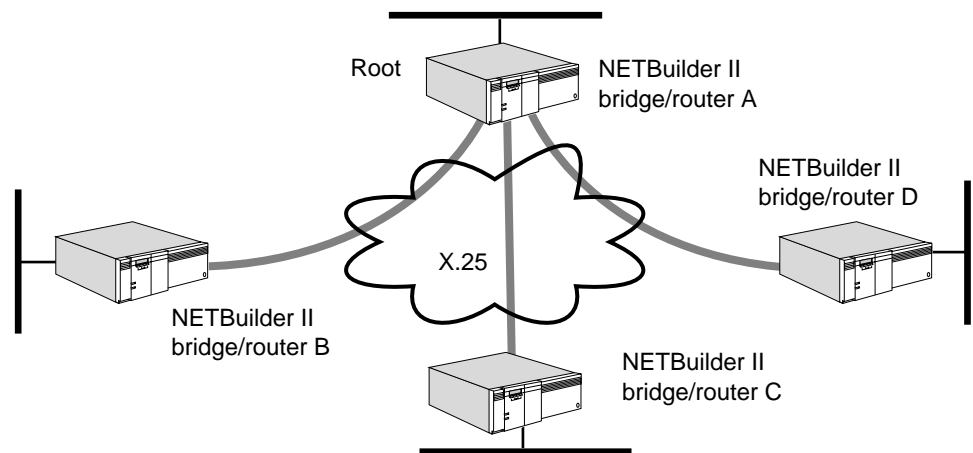
Fully Meshed, Partially Meshed, and Nonmeshed Topologies

A fully meshed X.25 topology is a topology where each node on a network is directly connected to all other nodes on the network. Each node is connected to the other nodes through a virtual circuit, and each virtual circuit has a DTE associated with it. Figure 389 shows an example of a fully meshed X.25 topology.

Figure 389 Fully Meshed X.25 Topology

The topology in Figure 389 consists of NETBuilder II bridge/routers. Through the established virtual circuits, bridge/router A is connected to bridge/routers B, C, and D; bridge/router B is connected to bridge/routers A, C, and D; and so on.

A nonmeshed X.25 topology is a topology where each node on a network is not necessarily connected to all other nodes on the network. Figure 390 shows an example of a nonmeshed X.25 topology.

Figure 390 Nonmeshed X.25 Topology

The topology in Figure 390 consists of NETBuilder II bridge/routers. Through the established virtual circuits, bridge/router A is connected to bridge/routers B, C, and D. bridge/routers B, C, and D are connected to bridge/router A only, but not to one another.

Two possible solutions exist to work around the lack of connectivity between bridge/routers B, C, and D. If you are routing IP-RIP, IPX, or AppleTalk, these protocols offer the next-hop split horizon feature. In IP-RIP, this feature is enabled when `-RIPIP CONTROL` is set to `NonMesh`. In IPX, it is enabled by manually

configuring neighbors. In AppleTalk, next-hop split horizon is configured by adding static mappings to the address mapping table.

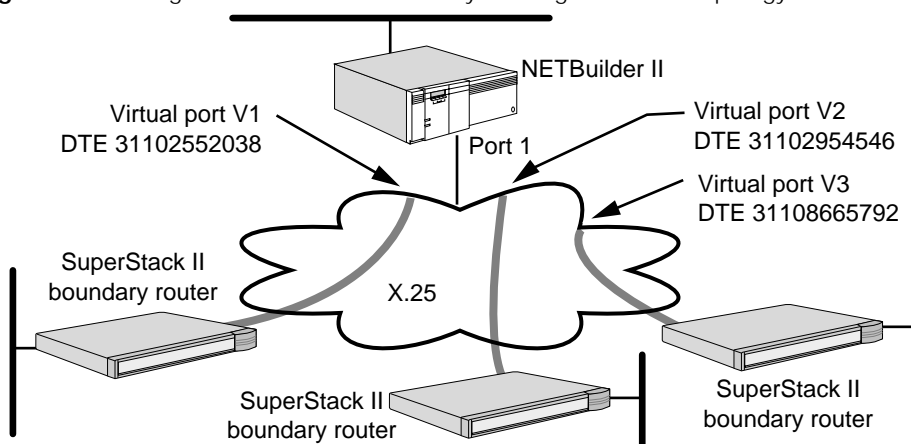
For example, if you are routing IP-RIP, use the SETDefault !<port> -RIP CONTROL = NonMesh syntax. If you are routing IPX, you can configure bridge/routers B, C, and D as neighbors using the PolicyControl and AdvToNeighbor parameters in the -NRIP and SAP Services. If routing AppleTalk, you can add the address of bridge/routers B, C, and D to an address mapping table. After taking such action, bridge/router A, the root bridge/router, learns available routes from each neighbor and then updates each neighbor with available routes other than that particular neighbor's own routes. Even though bridge/routers B, C, and D are not directly connected to one another, they can still learn of routes other than their own through bridge/router A. For more information on next-hop split horizon, see the Configuring AppleTalk Routing chapter, the Configuring IP Routing chapter, and the Configuring IPX Routing chapter.

Another solution in a topology where there is a lack of connectivity is to create virtual ports. Virtual ports are supported by bridging and all routing protocols over an X.25 network. You must use virtual ports in a Boundary Routing over X.25 topology and when bridging or routing DECnet, IP-OSPF, VINES, or XNS over X.25 in a partially meshed or nonmeshed topology. Using virtual ports in all other bridging or routing scenarios over an X.25 network is optional.

For information on the number of virtual ports supported per platform, see Table 11 in the Configuring Advanced Ports and Paths chapter.

Virtual ports allow the creation of multiple logical ports on one path. Each virtual circuit attaches a separate logical network. Figure 391 shows a Boundary Routing over X.25 topology where virtual ports are configured. In this topology, even though the SuperStack II NETBuilder boundary routers are not directly connected to one another, information about each of their networks can still be propagated through the NETBuilder II bridge/router.

Figure 391 Using Virtual Ports in a Boundary Routing Over X.25 Topology

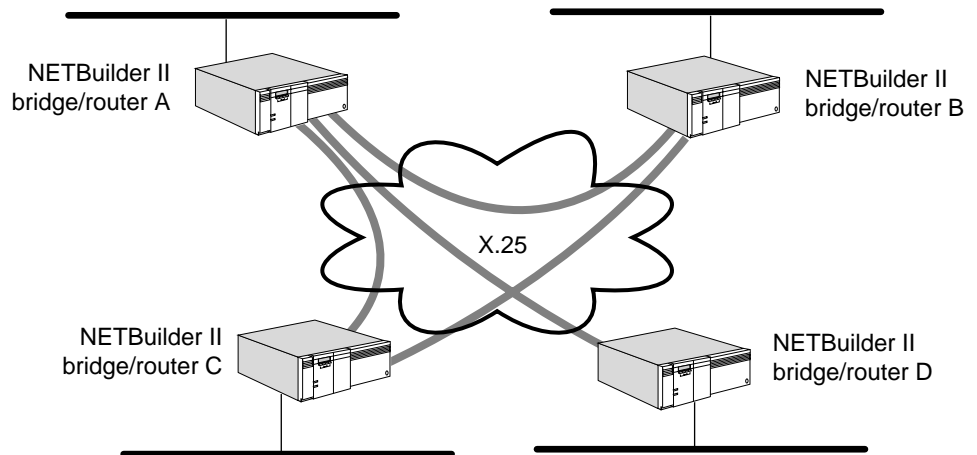


For more information on virtual ports and Boundary Routing over X.25, see the Configuring Advanced Ports and Paths chapter and the Configuring Boundary Routing System Architecture chapter, respectively.

A partially meshed X.25 topology is a topology where some nodes on a network are directly connected to all other nodes on the network (as in a fully meshed

topology) and other nodes are not (as in a nonmeshed topology). Figure 392 shows an example of a partially meshed X.25 topology.

Figure 392 Partially Meshed X.25 Topology



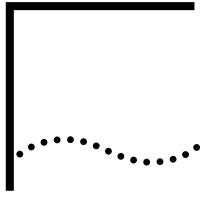
The topology in Figure 392 is composed of four NETBuilder II bridge/routers. Through the established virtual circuits, bridge/routers A, B, and C are connected to one another, but bridge/router D is connected to bridge/router A only.

The lack of connectivity between bridge/routers B, C, and D can be worked around using the same two solutions discussed earlier in this section that apply to nonmeshed topologies.

Facilities In addition to the basic X.25 functionality that is supported by all PDNs, another feature called *facilities* is optionally supported on some PDNs. Use of facilities is controlled at subscription time or on a call-by-call basis, depending on the facility.

The bridge/router supports the following facilities:

- Flow-control negotiation
- Throughput class negotiation
- Closed user group
- Fast select
- Fast select acceptance



CONFIGURING LOCAL AND GLOBAL SWITCHING

This chapter describes procedures for configuring the XSWitch Service on your bridge/router. The XSWitch Service consists of two features, local switching and global switching (X.25 tunneling over IP).

X.25 local switching allows the NETBuilder bridge/router to take an incoming call from a high-speed serial (HSS) port that is not targeted for the bridge/router itself and forward the call to its real X.25 destination by switching it over an X.25 WAN on another locally attached HSS port.

Global switching allows the bridge/router to take an incoming X.25 call that is not targeted for the bridge/router itself and, instead of switching the call to another HSS port, encapsulate and forward it through a locally attached IP Internet to another IP peer for further switching.

Switching can occur on either a *switched virtual circuit (SVC)* or a *permanent virtual circuit (PVC)*.

When configured for a switched virtual circuit and switching occurs, a *switched virtual circuit* is established. The switched virtual circuit is disconnected automatically when communication is complete.

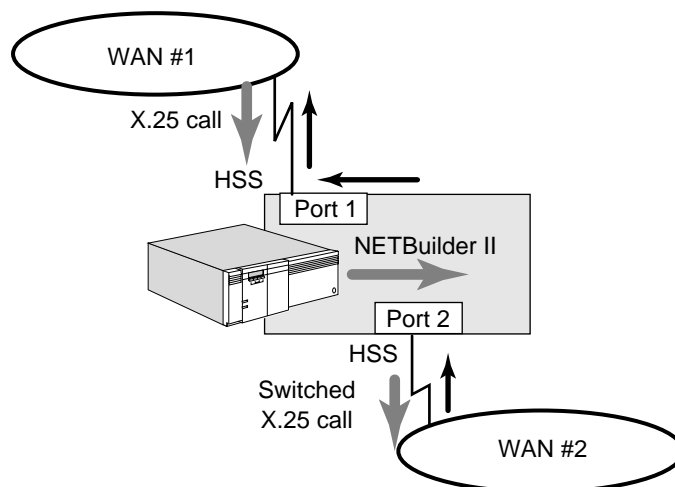
Using X.25 PVC support for tunneling, the circuit is maintained up at all times when the associated underlying interfaces are in the up state. When the PVC is properly configured and the NETBuilder bridge/router is booted, or when the HSS or LAN(IP) state is bounced, tunnel setup continuously attempts to connect the local-end to the remote-end until a tunnel circuit is established and running. The PVC tunnel is considered in the down state only when the HSS or LAN interface is in the down state.



For definitions of switching terms, see "Switching Terms" later in this chapter.

Setting Up Local Switching on a SVC

This section describes how to configure local switching on a switched virtual circuit. Figure 393 shows a bridge/router using local switching to forward an X.25 call from WAN #1 to WAN #2.

Figure 393 Local Switching

When the XSWitch Service receives an incoming X.25 call, it looks in the X25Prefix table to find an entry whose X.25 address prefix matches the address of the called address. When a match is found, its associated HSS port is used for switching. These X.25-prefix-to-HSS-port entries are user-configurable.

To configure local switching, follow these steps:

- 1 Verify that local switching is enabled by entering:

```
SHow -XSWitch CONTROL
```

If local switching is not enabled, enable it entering:

```
SETDefault -XSWitch CONTROL = Loc1SW
```

- 2 Assign X.25 prefix addresses to your HSS ports.

For example, to assign an X.25 prefix address of 5109 to port 2, enter:

```
ADD !2 -XSWitch X25Prefix 5109
```

For more information, see the XSWitch Service Parameters chapter in *Reference for Enterprise OS Software*.

Setting Up Global Switching on an SVC

This section describes how to configure global switching (X.25 tunneling over IP). Figure 394 shows an example of a bridge/router using tunneling to forward an X.25 call from WAN #1 to WAN #2.

Figure 394 Global Switching on an SVC

When the XSWitch Service receives an incoming X.25 call, it looks in the X25Prefix table to find an entry whose X.25 address prefix matches the address of the called address. When a match is found, its associated IP address is used for switching. These X.25-prefix-to-IP-address entries are user-configurable.

To configure global switching, follow these steps:

- 1 Verify that global switching is enabled by entering:

```
SHow -XSWitch CONTROL
```

If global switching is not enabled, enable it by entering:

```
SETDefault -XSwitch CONTROL = GlobSW
```

- Assign X.25 prefix addresses to your IP addresses.

For example, to assign an X.25 prefix address of 5109 to an IP address of 129.213.200.189, enter:

When a call is received with a prefix of 5109 it is mapped to the remote bridge/router via a tunnel.

```
ADD !129.213.200.189 -XSwitch X25Prefix 5109
```

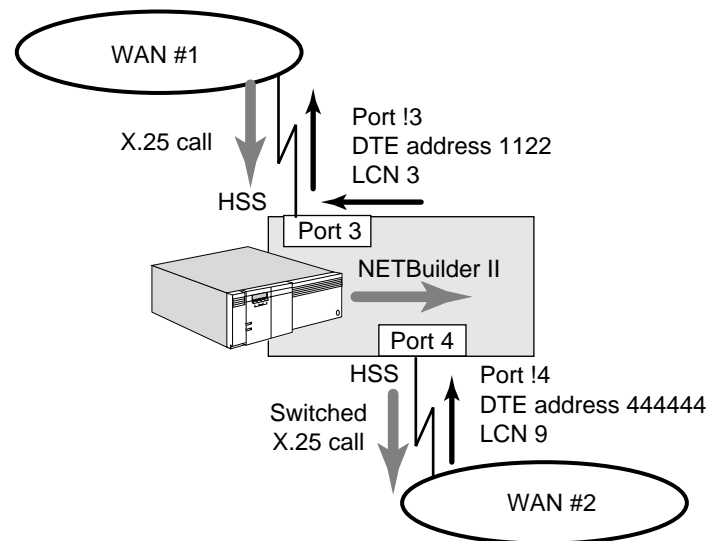
For additional parameters that affect global switching, see the XSwitch Service Parameters chapter in *Reference for Enterprise OS Software*.

Setting up Local Switching on a PVC

This section describes setting up local switching on a permanent virtual circuit.

Figure 395 is an example of using local switching on a PVC to forward an X.25 call from WAN #1 to WAN #2. This difference between local switching on and SVC and local switching on a PVC is the way in which the circuit is maintained.

Figure 395 Local Switching on a PVC



In local switching with PVCs, one router with two HSS ports is involved for each switched circuit. The configuration requires on XSWPVC to indicate an incoming PVD and the switched outgoing PVC mapping. As in global switching circuits, the local switching PVC circuit should stay up and running as long as the router is operating and both HSS ports are in the UP state.

To configure local switching on a permanent virtual circuit, follow these steps:

- Configure the permanent virtual circuits by entering:

```
ADD !3 -X25 PVC 3,3 1122 FF 0
ADD !4 -X25 PVC 9,9 444444 FF 0
```

These commands create PVC connections on ports 3 and 4. These PVCs carry switched traffic as specified by the protocol ID FF, to and from logical channel numbers 3 and 9 with DTE addresses 1122 and 444444 respectively.



Always use protocol identifier FF to indicated switched PVCs.

- 2 To verify the X.25 PVC configuration, enter:

SHoW -X25 PVC

A display similar to the following appears:

```
Port !3 PVC 3,3 1122 FF 0
Port !4 PVC 9,9 444444 FF 0
```

- 3 Specify the tunnel by entering:

ADD !3 -XSwitch XSWPVC 1122 3 !4 444444 9

This command maps a circuit from port 3 with DTE address 1122 and logical channel number 3 into the target destination DTE address 444444 and logical channel number 9 which is port 4.

- 4 To verify the configuration, enter:

SHoW -XSwitch XSWPVC

A display similar to the following appears:

| Port#/IPAddr | SDTE | SLCN | DESTPort/#IPAddr | DDTE | DLCN |
|--------------|------|------|------------------|-------|------|
| !3 | 1122 | 3 | !4 | 44444 | 9 |
| | | | | 4 | |

This display shows that a PVC from source port 3 with DTE address 1122 will be switched to destination port 4 with DTE address 444444 and local channel number 9.

- 5 To verify that a locally switched X25 PVC is up and running enter:

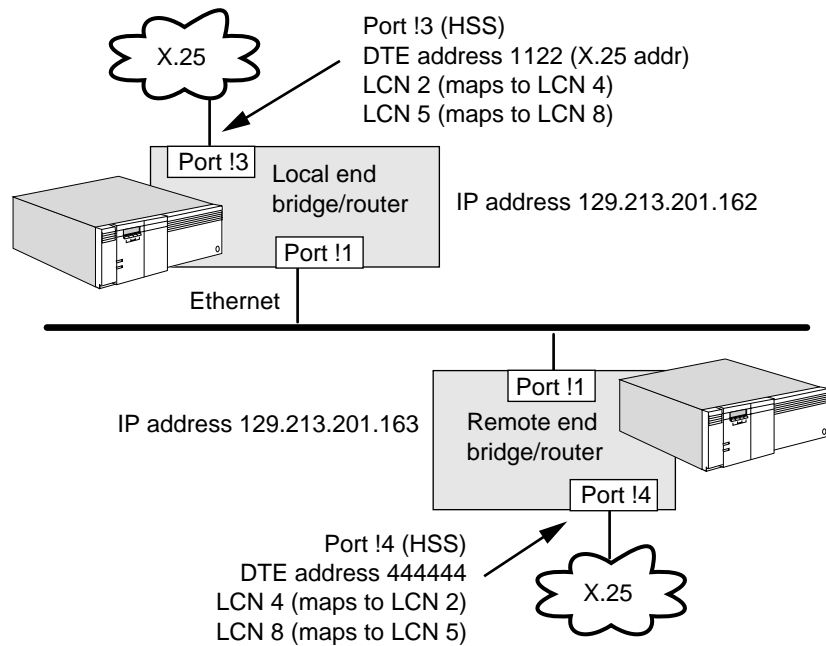
SHOW -XSwitch SwitchedVC

A display similar to the following appears.

```
SW# XSRC SDST SRC(LCN) DST(LCN)STATE BYTESXFER
0 1122 444444 !3(4) !4(9) ACT 0*
* Indicates X25 in the switch circuit.
```

Setting up Global Switching on a PVC

This section describes how to configure global switching (X.25 tunneling over IP). Figure 394 shows an example of a bridge/router using tunneling to forward an X.25 call from WAN #1 to WAN #2 on a permanent virtual circuit.

Figure 396 Global Switching on a Permanent Virtual Circuit over a LAN

A tunnel is established between two NETBuilder bridge/routers with one bridge/router acting as the local end and the other acting as the remote end. Multiple circuits can be supported between two NETBuilder bridge/routers where each circuit is set up independently.

The local end (source) and remote end (destination) addresses can be an ip address or HSS port. For tunnel mapping, one address must be an HSS port and the other must be an ip address. When the local-end (source) is an HSS port and the remote-end (destination) is an ip address, the circuit is called a local-end of the tunnel. When the local-end (source) is an ip address and the remote-end is an HSS port, the tunnel is called a remote-end tunnel. The NETBuilder bridge/router can support both local-end and remote-end of the tunnels at the same time as long as each circuit is properly configured on both NETBuilder bridge/routers.

Using X.25 PVC support for tunneling, the circuit is maintained up when the underlying associated interfaces are in the up state. When the PVC is properly configured and the NETBuilder bridge/router is booted, or when the HSS or LAN(IP) state is bounced, tunnel setup continuously attempts to connect the local-end to the remote-end until a tunnel circuit is established and running. The PVC tunnel is considered in the down state only when the HSS or LAN interface is in the down state.

Configuring the Local-end Router

This example shows how to configure two PVC switch circuits in a tunnel. To configure global switching on a permanent virtual circuit, on the local-end NETBuilder bridge/router, follow these steps:

- 1 To specify the permanent virtual circuit with a profile ID (FF) set to switching, enter:

```
ADD !3 -X25 PVC 2,2 1122 FF
ADD !3 -X25 PVC 5,5 1122 FF
```

These commands indicate that logical channel numbers 5 and 2 from port !3 with the DTE address 1122 will be switched.

- 2 Verify that the PVC is properly configured by entering:

```
SHow -X25 PVC
```

A display similar to the following should appear:

```
Port !3 PVC 5,5 1122 FF 0
Port !3 PVC 2,2 1122 FF 0
```

These two entries indicate that logical channel numbers 5 and 2 from port !3 will with DTE address 1122 will be switched.

- 3 To specify the tunnel, enter:

```
ADD !3 -XSwitch XSWPVC 1122 2 129.213.201.163 444444 4
ADD !3 -XSwitch XSWPVC 1122 5 129.213.201.163 444444 8
```

The first command maps a circuit from port 3, DTE #1122, logical channel number 2 into a remote end via tunnel into 129.213.201.163 with a final destination of DTE#444444, logical channel number 4. The second command maps a circuit from port!3, DTE #1122, logical channel number 5 into a remote end via tunnel into 129.213.201.163 with a final destination of DTE#444444, logical channel number 8.

- 4 Verify that the tunnel is configured properly by entering:

```
SHow -XSwitch XSWPVC
```

A display similar to the following should appear:

```
Port#/IPAddrSDTESLCNDESTPort/IPAddr DDTE DLCN
!3 11222 129.213.201.163444444 4
!3 11225 129.213.201.163444444 8
```

Entry number one maps a circuit from port 3 with DTE#1122 and logical channel number 2 into a remote tunnel with its final destination as DTE #444444 with logical channel number 4. Entry number two maps a circuit from port 3 with DTE address 1122 and logical channel number 5 to its final destination at DTE address 444444 with logical channel number 8

Configuring the Remote-end Router

To configure global switching on a permanent virtual circuit, on the remote-end NETBuilder bridge/router, follow these steps:

- 1 To specify the permanent virtual circuit with a profile ID (FF) set to switching, enter:

```
ADD !4 -X25 PVC 8,8 444444 FF
ADD !4 -X25 PVC 4,4 444444 FF
```

These commands indicate that logical channel numbers 8 and 4 from port 4 with the DTE address 444444 will be switched.

- 2 Verify that the PVC is properly configured by entering:

```
SHow -X25 PVC
```

A display similar to the following should appear:

```
Port !4 PVC 8,8 444444 FF 0
Port !4 PVC 4,4 444444 FF 0
```

These two entries indicate that logical channel numbers 8 and 4 from port 4 will with DTE address 444444 will be switched.

- 3 To specify the tunnel, enter:

```
ADD !129.213.201.162 -XSwitch XSWPVC 1122 2 !4 444444 4
ADD !129.213.201.162 -XSwitch XSWPVC 1122 5 !4 444444 8
```


The first command maps a circuit from ip address 129.213.201.162 with the DTE source DTE#1122, logical channel 2 into its destination via HSS port 4 with local channel 4 and DTE address 444444. The second command maps a circuit from ip address 129.213.201.162 to the DTE#1122, logical channel number 5 into its destination via HSS port 4 with local channel number 8 and DTE. address 444444.

- 4 Verify that the tunnel is configured properly by entering:

```
SHoW -XSWitch XSWPVC
```

A display similar to the following should appear:

| Port/IPAddr | SDTE | SLCN | DEST | Port/IPAddr | DDTE | DLCN |
|----------------|------|------|------|-------------|--------|------|
| 129.21.201.162 | 1122 | 2 | !4 | 444444 | 4 | |
| 129.21.201.162 | 1122 | 5 | | !4 | 444444 | 8 |

Entry one shows that a tunnel is mapped from 129.213.201.162 with the DTE address of DTE#1122 and logical channel number 2 into its destination via the HSS port !4, with logical channel number 4 and DET address DTE#444444.

Entry two shows that a tunnel is mapped from 129.213.201.162 with the DTE address of DTE address 1122 and logical channel number 5 into its destination via the HSS port 4, with logical channel number 8 and DET address DTE address 444444.

- 5 Verify that the tunnel X25 PVC is up and running by entering:

```
ShoW -XSWitch SWitchedVC
```

A display similar to the following appears:

```
SW# XSRC SDST SRC(LCN) DST (LCN) STATE BYTESXFER
0 1122 444444 129.213.201.162 !4(4) ACT 0*
1 1122 444444 129.231.201.162 !4(8) ACT 0*
* Indicates X25 in the switch circuit.
```

When correctly configured the bridge/routers local and remote will attempt to set up a tunnel between each other automatically. Automatic setup should also occur when the port is bounced (port down and then back up again).

If this is the first time configuration for the router, you may need to toggle the path/port to start the PVC tunnel set-up sequence.

A typical error occurs when the two ends of the tunnel have a mismatch in the XSWPVC values. When is mismatch occurs the tunnel will not set up properly. When the router detects this configuration error it will report the following messages:

```
WARNING: A misconfiguration of PVC or XSWPVC!!!
Please: Correct the configuration and
DElete -XSWitch SWitchedVC ALL on both sides.
```

When this message is displayed, follow these recovery steps:

- 1 Verify with your network diagram, and check to see if the configuration setup for PVC and XSWPVC are matched on both ends of the tunnel. On both the local and the remote routers, enter:

```
SHoW -X25 PVC
SHoW -XSWitch XSWPVC
```

Correct parameters as required.

- 2 Bounce (toggle) the HSS port by disabling the path and then re-enabling the path.

- 3 Verify that the setup is correct by entering:

```
SHow -XSwitch SwitchedVC
```

A display similar to the following appears:

```
SW# XSRC SDST SRC (LCN) DST (LCN) STATE BYTESXFER
0 1122 5555555 !3 (2) 10.11.12.14. ACT 168 *
* Indicates and X25 PVC in the switch circuit.
```

The state ACT indicates that the tunnel is in Active State.

The Bytesxfer field reports the number of bytes of data traveling through this circuit.

Setting up Switching on a PVC Over a WAN

This section describes how to configure global switching over WAN media. Figure 394 shows an example of a bridge/router using tunneling to forward multiple X.25 calls from WAN #1 to WAN #2, WAN #3 and WAN #4 on a permanent virtual circuit.

Figure 397 Global Switching on a PVC over a WAN

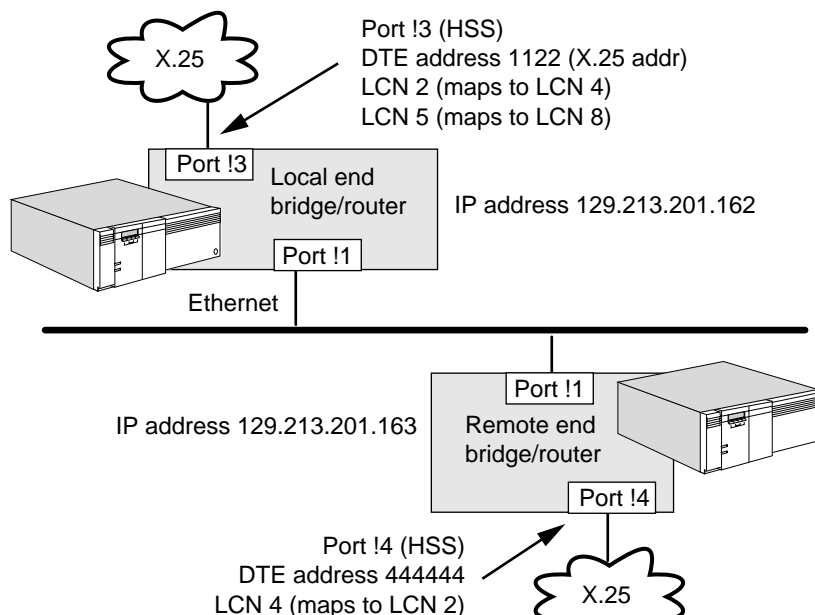


Figure 397 is an example of setting up NETBuilder bridge/routers to use tunnelled PVCs to other routers. In this example, the HSS port used on each router is port 4. Also, the user profile profile identifier 0 is used. For each PVC a fake DTE address is created to associate with the PVC to identify the local end and the remote end of the tunnel. 1111, 2222, 3333, and 4444 are fake ids. One fake DTE address can associate may PVC. For example, 1111 in bridge/router associates with its local logical channel numbers 1, 2, and 3.

Configuring Local Router A

To configure bridge/router A, follow these steps:

- 1 Configure -X25 PVC for logical channel numbers 1, 2, and 3 by entering:

```
ADD !4 -X25 PVC 1,1 1111 FF 0
ADD !4 -X25 PVC 2,2 1111 FF 0
ADD !4 -X25 PVC 3,3 1111 FF 0
```

These commands add permanent virtual circuits to HSS port 4. Associates logical channel number 1, 2 and 3 with the fake DTE address 1111, indicates Switching with protocol identifier FF and establishes the user profile id as 0.

- 2 Configure the -XSwitch service XSWPVC parameter for logical channel numbers 1, 2, and 3 by entering:

```
ADD !4 -XSwitch XSWPVC 1111 1 128.102.100.100 2222 1
ADD !4 -XSwitch XSWPVC 1111 2 128.102.100.101 3333 1
ADD !4 -XSwitch XSWPVC 1111 3 128.102.100.103 4444 1
```

The first command establishes a tunnel with bridge/router B in the example configuration. !4 indicates that the incoming HSS port is 4. 1111 is the associated fake DTE address, the first 1 is the logical channel 1 on the source side, 128.102.100.100 is the target tunnel ip address (router B), 2222 is the target fake DTE address; the final 1 is the logical channel number 1 at the target end (router B.)

The second and third commands establish similar settings for the other two routers in the example configuration.

Configuring the Remote Routers

Next the target ends of the tunnels need to be configured on the remote routers.

Configuring Remote Router B

To configure remote bridge/router B, follow these steps:

- 1 Configure the -X25 PVC by entering:

```
ADD !4 -X25 PVC 1,1 2222 FF 0
```

This command specifies port 4 as the HSS port, 1,1 indicates the pvc_range which is logical channel number 1 on the router B side; 2222 is the fake DTE address, FF is the protocol identifier indicating switching, 0 is the user profile identifier.

- 2 Configure the -XSwitch service XSWPVC parameter by entering:

```
Add !128.102.100.102 -XSwitch XSWPVC 1111 1 !4 2222 1
```

This command establishes 128.102.100.102 as the incoming tunnel address which is in this case router A. 1111 is the source DTE address which is in this case router A, The first 1 indicates the logical channel number 1 on router A. The HSS port 4 means the outgoing HSS port on router B. 2222 is the fake DTE address and the last 1 is the destination logical channel number on router B.

Configuring Remote Router C

To configure router C, follow these steps:

- 1 Configure the -X25 PVC by entering:

```
ADD !4 -X25 PVC 1,1 3333 FF 0
```

This command specifies port 4 as the HSS port, 1,1 indicates the PVC range which is logical channel number 1 on the router C side; 3333 is the fake DTE address, FF is the protocol identifier indicating switching, 0 is the user profile identifier.

- 2 Configure the -XSWitch XSWPVC parameter by entering:

```
Add !128.102.100.102 -XSWitch XSWPVC 1111 2 !4 2222 1
```

This command establishes 128.102.100.102 as the incoming tunnel address which is in this case router A. 1111 is the source DTE address which is in this case router A, The 2 indicates the logical channel number 2 on router A. The HSS port 4 means the outgoing HSS port on router C. 3333 is the fake DTE address and the last 1 is the destination logical channel number on router C.

Configuring Remote Router D

To configure router D, follow these steps:

- 1 Configure the -X25 PVC by entering:

```
ADD !4 -X25 PVC 1,1 4444 FF 0
```

This command specifies port 4 as the HSS port, 1,1 indicates the pvc_range which is logical channel number 1 on the router D side; 4444 is the fake DTE address, FF is the protocol identifier indicating switching, 0 is the user profile identifier.

- 2 Configure the -XSWitch XSWPVC parameter by entering:

```
Add !128.102.100.102 -XSWitch XSWPVC 1111 3 !4 4444 1
```

This command establishes 128.102.100.102 as the incoming tunnel address which is in this case router A. 1111 is the source DTE address which is in this case router A, The 3 indicates the logical channel number 3 on router A. The HSS port 4 means the outgoing HSS port on router D. 4444 is the fake DTE address and the last 1 is the destination logical channel number on router D.

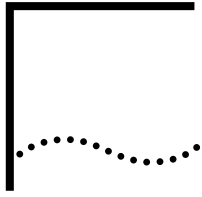


There are several parameters that need to be adjusted based on how this configuration is established. You may need to configure X.25, Level 2, and Level 3 parameters to match the values in the entered in this procedures. See the values for the parameters in the PATH Service, the LAPB Service, the X25 Service, the PORT Service and the PROFILE Service.

Switching Terms

The following terms are used in this chapter to explain switching:

| | |
|-------------------|---|
| tunneling service | A method of connecting peer internets that are not physically reachable with the X.25 Protocol. This is a generic service on NETBuilder bridge/routers. Global switching interfaces with it to set up and maintain the tunnel between two entities over the Internet. |
| encapsulation | Conveying an X.25 packet within a TCP data packet so it can be forwarded through a TCP connection. |
| decapsulation | Extracting an X.25 packet encapsulated in a TCP data packet for further forwarding through a locally attached X.25 WAN. |
| Local-end tunnel | For tunnel mapping, one address must be an HSS port and the other must be an ip address. When the local-end (source) is an HSS port and the remote-end (destination) is an ip address, the circuit is called a local-end tunnel. |
| Remote-end tunnel | When the local-end (source) is an ip address and the remote-end is an HSS port, the tunnel is called a remote-end tunnel. |



CONFIGURING CONNECTIONS FOR OUTGOING CALLS

This chapter describes how to configure your bridge/router to function as an X.25 connection service gateway for outgoing calls. The gateway allows end users to make connections from IP Internet-attached Telnet clients, raw Transmission Control Protocol (TCP) clients, and Open Systems Interconnection (OSI) Virtual Terminal Protocol (VTP) clients to X.25-attached hosts that support the X.29 Protocol. Procedures in this chapter include how to make outgoing automatic (one-step) and extended (two-step) connections.



The NETBuilder II bridge/router supports 128 connection service sessions.



For conceptual information, see "How the Outgoing Connection Service Works" later in this chapter.

Setting Up the Gateway for Outgoing Telnet Connections

This section describes how to configure the bridge/router gateway to handle outgoing connections for Telnet clients.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the system's local and wide area ports and paths according to the procedure in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the X.25 interface.

After completing the procedure for local area and wide area paths and ports, make sure you configure X.25 as the owner of each wide area interface to be used in the outgoing connection service using:

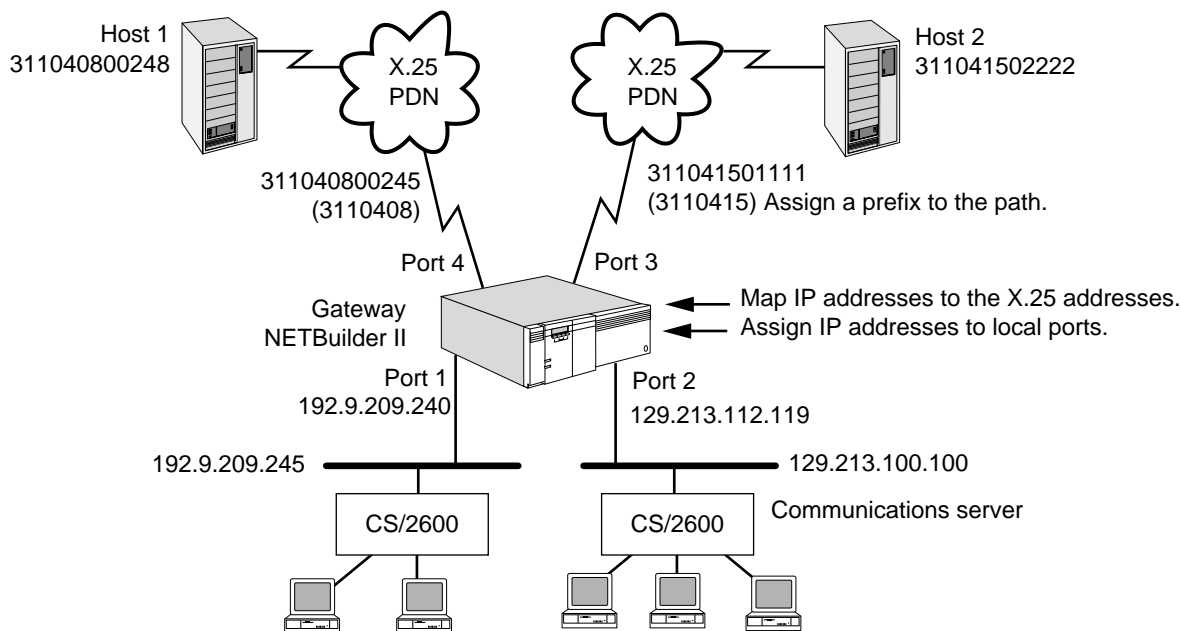
```
SETDefault !<port> -PORT OWNEr = X25
```

To configure the X.25 interface, see the Configuring Wide Area Networking Using X.25 chapter.

Before making outgoing automatic or extended connections, you must provide a list of connection addresses as well as connection and disconnection commands to end users of the connection service gateway. For information on making outgoing connections, see "Making Outgoing Connections" later in this chapter.

Procedure

After configuring ports, paths, and the X.25 interface, you need to configure the gateway for outgoing automatic and extended Telnet connections. Use Figure 398 and the following procedure to configure the gateway.

Figure 398 Connection Service Configuration Overview (Telnet)

To configure the gateway for outgoing automatic and extended Telnet connections, follow these steps:

- 1 Before configuring the gateway for outgoing Telnet connections, display address information for directly connected IP networks by entering:

```
SHoW -IP NETaddr
```

The following display appears:

```
-----IP Directly Connected Networks-----
IP Address      Port  Subnet Mask      Status  MTU  Broadcast  Format
129.213.112.119  2    255.255.255.0    Up      1500 129.213.112.255
```

You need to configure an IP address for each Ethernet interface used for connection service purposes. For example, to add the address 192.9.209.204 for port 1, enter:

```
SETDefault !1 -IP NETaddr = 192.9.209.240 255.255.255.0
```



For the gateway to accept a raw TCP connection, you need to add a listen port using the `-TCPAPPL LISTenerPorts` parameter.

- 2 Create a table that maps assigned IP addresses to X.25 addresses of the hosts to which users will want to connect using:

```
ADD [!<configfile>] -Gateway IPX25Map <IPaddress> {X25 addr string} | PAD}
```

- a Select a configuration file, if necessary.

When creating the table, you can specify a configuration file to initialize the port and session before outgoing connections are made. If you do not specify a configuration file, then configuration file 2 is used as the default. You can usually use configuration file 2 without modification; the default settings of the TERM Service parameters are acceptable for most outgoing connections. If you require different settings than the defaults already provided, use one of the configuration files numbered 3 through 32.

Configuration file 1 is the default for incoming connections and must not be used for outgoing connections. If you use an odd-numbered configuration file for an outgoing connection, make sure you change the DeVice parameter from Terminal to Host by entering the SETDefault !configfile -TERM DeVice = Host command, or the connection attempt will fail.

For information on TERM Service parameters specifically needed for outgoing connections, see the TERM Service Parameters chapter in *Reference for Enterprise OS Software*. For information on how to map TERM Service parameters to X.3 parameters for outgoing connections, see the X.3 Parameters and PAD Profiles appendix.

- b** Configure an IP address that is on the same network or subnetwork to which the gateway is attached.

An IP address assigned to an X.25 address for establishing an automatic outgoing connection must be valid on some IP subnet to which the gateway is attached. For example, if the gateway has two LAN ports and is configured to route IP packets between these two ports, the gateway will be attached to two IP subnets, and an IP address assigned to an IPX25Map entry must be derived from one of these subnets.

For example, the IP address for port 1 is 192.9.209.240. An IP address with a subnet mask of 255.255.255.0 used with the IPX25Map parameter can have the network portion 192.9.209; you can assign the host portion of the address a subnet between 1 and 254 that has not already been assigned. Similarly, the IP address for port 2 is 129.213.112.119 with a subnet mask of 255.255.255.0, and the network portion is 129.213.112; you can assign the host portion of the address a subnet between 1 and 254 that has not already been assigned. For information about Internet addressing, see the Internet Addressing appendix.

- c** Select either an X.25 information string or the keyword PAD.

The IPX25Map parameter requires that an X.25 address string (used for automatic connections) or the keyword PAD (used for extended connections) follow the IP address. Table 93 summarizes the X.25 address strings that can be used with the IPX25Map parameter.

Table 93 X.25 Information Strings for Automatic Connections

| To Specify | Options | X.25 Address String Example* |
|------------------------------|---|--------------------------------------|
| X.25 host information | No options | 311040800248 |
| | Call user data: | |
| | Display the data string on the destination terminal | 311040800248DHELLO† |
| | Hide (protect) the string on the destination terminal | 311040800248PHELLO† |
| | Facilities: | |
| | Reverse charge request (R- or R* can be used) | R-311040800248
R*311040800248 |
| | Closed user group (G09- or G09* can be used) | G09-311040800248
G09*311040800248 |

Table 93 X.25 Information Strings for Automatic Connections (continued)

| To Specify | Options | X.25 Address String Example* |
|---------------------|---|--|
| | Reverse charge request and closed user group (R, G09- or R, G09* can be used) | R,G09-311040800248
R,G09*311040800248 |
| (continued) | | |
| | Facilities and call user data: | To:Debra Knodel/HQ/3Com
cc:Patrick Sullivan/HQ/3Com
To:Debra Knodel/HQ/3Com
cc:Patrick Sullivan/HQ/3Com |
| | Reverse charge request and call user data | R*311040800248DHELLO |
| | Closed user group and call user data | G09*311040800248PHELLO |
| | Reverse charge, closed user group, and call user data | R,
G09-311040800248DHELLO |
| Private line | No options | L (The gateway uses the lowest path enabled for connection service.) |
| | A path | L3 (directly selects line 3) |
| | A path and call user data | L4DHELLO (directly uses line 4 with call user data) |

* P, D, R, G, L and call user data can be entered in upper- or lowercase.

† P and D distinguish the end of the X.25 address from the call user data.

d Configure address mappings for automatic connections.

The following three examples show IP-to-X.25 address mappings for automatic connections:

- To map the IP address 192.9.209.100 to the X.25 host address 311040800248 with reverse charging and to initialize the port and session with configuration file 2, enter:

```
ADD -Gateway IPX25Map 192.9.209.100 R*311040800248
```

- If no configuration file is specified in the command, as shown in this example, the gateway automatically uses configuration file 2 as the default.
- To map the IP address 192.9.209.101 to the X.25 host address 311041502222 with reverse charging and closed user group facilities, and to initialize the port and session with configuration file 4, enter:

```
ADD !4 -Gateway IPX25Map 192.9.209.101 R,G09*311041502222
```

- To map the IP address 129.213.112.120 to the X.25 host address 311041502222 with reverse charging and to send call user data, and to initialize the port and session with configuration file 4, enter:

```
ADD !4 -Gateway IPX25Map 129.213.112.120 R-311041502222DHELLO
```

With automatic connections, the gateway automatically places a call to the destination X.25 host address, supports the named facility, reverse charge and/or closed user group, and sends call user data as requested. For more information, see "Making Outgoing Connections" and "Automatic Connections" later in this chapter.

e Configure address mappings for extended connections.

The following example shows how to make an IP-to-X.25 address map for an extended connection. To place the caller who makes a connection request into PAD emulation mode and to initialize the port and session with configuration file 4, enter the PAD keyword after the IP address as follows:

```
ADD !4 -Gateway IPX25Map 129.213.112.121 PAD
```

PAD emulation mode allows the caller to select a different profile, alter PAD parameters, and establish virtual calls to X.25 hosts. For information on PAD emulation mode, see "Making Outgoing Connections" and "Extended Connections" later in this chapter.

- f Display the IP address-to-X.25 address mappings by entering:

```
SHow -Gateway IPX25Map
```

The following display appears:

| Config File | IP Address | X.25 Information |
|-------------|-----------------|----------------------|
| ! 2 | 192.9.209.100 | R*311040800248 |
| ! 4 | 192.9.209.101 | R,G09*311041502222 |
| ! 4 | 129.213.112.120 | R-311041502222DHELLO |
| ! 4 | 129.213.112.121 | (PAD mode) |

- 3 Configure the X25Prefix parameter in the XSWitch Service so that the gateway can select the wide area path for reaching the X.25 host.

Make sure to configure this parameter for each wide area path that is enabled for connection service.

The prefix is a series of numbers that match the destination X.25 address in part or completely. For example, to configure the gateway's wide area port 3 with a prefix that identifies the X.25 host with address 311041502222, enter:

```
ADD !3 -XSWitch X25Prefix 3110415
```

When a connection request is made, the gateway scans the prefix table for a prefix that matches the target X.25 address. If no match exists for the target X.25 address, the connection request is denied. To prevent a connection denial because of no matching prefix, you can select one default port for the gateway. For example, you can select port 4 to be the default port by using the default option in the following command:

```
ADD !4 -XSWitch X25Prefix Default
```

To display the prefix-to-port mapping, enter:

```
SHow -XSWitch X25Prefix
```

The following display appears:

| Port #/IPAddr | X.25 Prefix |
|---------------|-------------|
| !4 | Default |
| !3 | 3110415 |
| !4 | 3110408 |

- 4 Verify that the gateway paths are configured for outgoing automatic and extended connections.

For example, to verify path 3 is configured, enter:

```
SHow !3 -Gateway CONTROL
```

The following display appears:

```
Path !3 CONTrol = (Enable, InExt, OutExt, InAuto, OutAuto, DDXP,
  SubAddr, NoDSA, NoTrace)
```

By default, the CONTrol parameter is disabled; however, it automatically becomes enabled when the X.25 path comes up.

If the display is incorrect, you can configure the path for the type of operation desired. For example, to enable the gateway on wide area path 3 for both outgoing automatic and extended connections, enter:

```
SETDefault !3 -Gateway CONTrol = (Enable, OutExt, OutAuto)
```

After configuring the X25 Service and Gateway Service parameters, see the next section for information on making outgoing automatic and extended connections.

Setting Up the Gateway for Outgoing VTP Connections

This section describes how to configure the bridge/router gateway to handle outgoing connections for OSI VTP clients.

Prerequisites

Before beginning this procedure, complete the following steps:

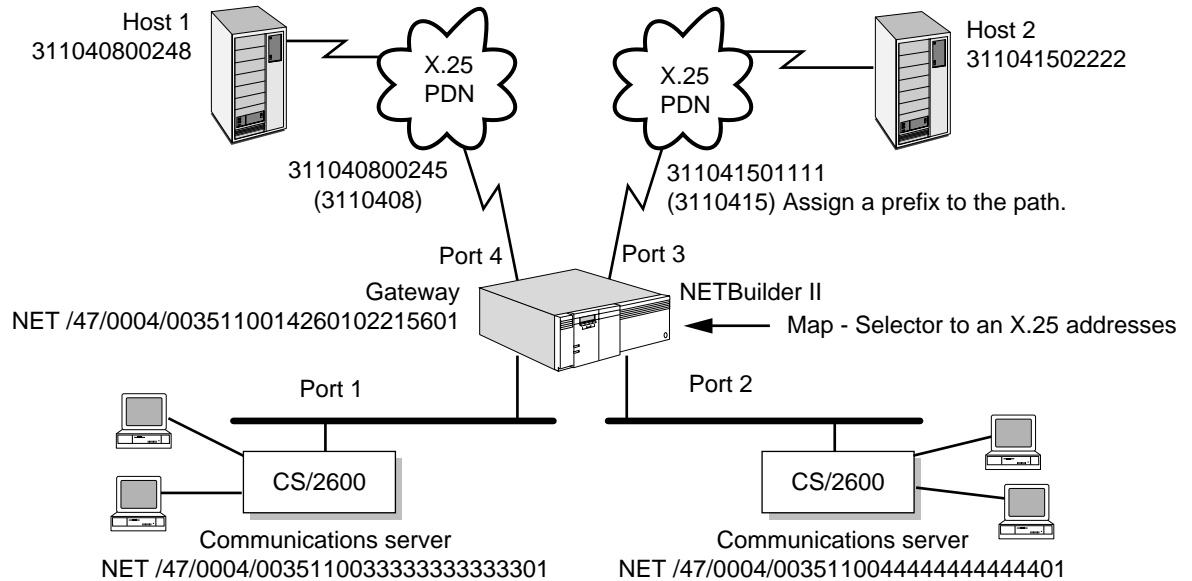
- Log on to the system with Network Manager privilege.
- Set up the local and wide area ports and paths using the procedure in the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the X.25 interface.

After completing the procedure for local area and wide area paths and ports, make sure you configure X.25 as the owner of each wide area interface to be used in the outgoing connection service. For example, for each wide area interface use:

```
SETDefault !<port> -PORT OWNeR = X25
```

Procedure

After configuring ports, paths, and the X.25 interface, you need to configure the gateway for outgoing automatic and extended VTP (OSI) connections. Use Figure 399 and the following procedure to configure the gateway.

Figure 399 Connection Service Configuration Overview (OSI)

To configure the gateway for outgoing automatic and extended VTP (OSI) connections, follow these steps:

- 1 Before configuring the gateway for outgoing OSI connections, display the Network Entity Title (NET) by entering:

```
SHoW -CLNP NetEntityTitle
```

The following display appears:

```
NetEntityTitle = /49/005308000212345600
```

For more information about configuring the NetEntityTitle, see "AreaAddress" in the ISIS Service Parameters chapter in *Reference for Enterprise OS Software*.

- 2 Create a table that maps assigned P-Selector to X.25 addresses of the host to which users will want to connect.

Use:

```
SETDefault !<P-Sel> -Gateway PselX25Map {![<config file>] (<x.25 addr string> | PAD ) | None}
```

The P-Selector in the Presentation Address must be 2 octets in length and the value of the first octet must be 0. When you want to make a connection using Telnet profiles, and 4 is used, 0 or 4 is used for the X.3 profiles. As a result, the mapping is only for the second octet of the P-Selector. The X.3 profile can only be applied for outgoing automatic connections.

- a If you are using Telnet profiles, select a configuration file if necessary.

When creating the table, you can specify a configuration file to initialize the port and session before outgoing connections are made. If you do not specify a configuration file, then configuration file 2 is used as the default. You can usually use configuration file 2 without modification; the default settings of the TERM Service parameters are acceptable for most outgoing connections. If you require different settings from the defaults already provided, use one of the configuration files numbered 3 through 32.

Configuration file 1 is the default for incoming connections and must not be used for outgoing connections. If you use an odd-numbered configuration file for an outgoing connection, make sure you change the DeVice parameter from Terminal to Host by entering the SETDefault !configfile -TERM DeVice = Host command, or the connection attempt will fail.

For information on TERM Service parameters specifically needed for outgoing connections, see the TERM Service Parameters chapter in *Reference for Enterprise OS Software*. For information on how to map TERM Service parameters to X.3 parameters for outgoing connections, see the X.3 Parameters and PAD Profiles appendix.

- b** Configure the second octet of the P-Selector to map to an X.25 address.

Use:

```
SETDefault !<P-Sel> -Gateway PSelX25Map
```

Select either an X.25 address string or the keyword PAD.

The PSelX25Map parameter requires either an X.25 address string (used for automatic connections) or the keyword PAD (used for extended connections). For example, to map P-Selector 4 to the X.25 address 311040800248, enter:

```
SETDefault !4 -Gateway PSelX25Map = 311040800248
```

To set P-Selector 2 to PAD for extended connections, enter:

```
SETDefault !2 -Gateway PSelX25Map = PAD
```

Table 93 summarizes the X.25 address strings that can be used with the PSelX25Map parameter.

- c** To display the P-Selector-to-X.25 address mappings, enter:

```
SHow -Gateway PSelX25Map
```

The following display appears:

| Config File | P-Selector | X.25 Information |
|-------------|------------|------------------|
| !2 | 01 | 311040800248 |
| !2 | 02 | (PAD Mode) |

- 3** Configure the X25Prefix parameter in the XSWitch Service so that the gateway can select the wide area path for reaching the X.25 host.

Make sure you configure this parameter for each wide area path that is enabled for connection service.

The prefix is a series of numbers that match the destination X.25 address in part or completely. For example, to configure the gateway's wide area port 3 with a prefix that identifies the X.25 host with address 311041502222, enter:

```
ADD !3 -XSWitch X25Prefix 3110415
```

When a connection request is made, the gateway scans the prefix table for a prefix that matches the target X.25 address. If no match exists for the target X.25 address, the connection request is denied. To prevent a connection denial because of no matching prefix, you can select one default port for the gateway. For example, you can select port 4 to be the default port by using the Default option when you enter:

```
ADD !4 -XSWitch X25Prefix Default
```

To display the prefix-to-port mapping, enter:

```
SHow -XSwitch X25Prefix
```

The following display appears:

```
Port #/IPAddr          X.25 Prefix
-----
!4                      Default
!3                      3110415
!4                      3110408
```

- 4 Verify that the gateway paths are configured for outgoing automatic and extended connections.

For example, to verify path 3 is configured, enter:

```
SHow !3 -Gateway CONTrol
```

The following display appears:

```
Path !3 CONTrol = (Enable, InExt, OutExt, InAuto, OutAuto, DDXP,
SubAddr, NoDSA, NoTrace)
```

By default, the CONTrol parameter is disabled. However, it automatically becomes enabled when the X.25 path comes up.

If the display is incorrect, you can configure the path for the type of operation desired. For example, to enable the gateway on wide area path 3 for both outgoing automatic and extended connections, enter:

```
SETDefault !3 -Gateway CONTrol = (Enable, OutExt, OutAuto)
```

The gateway also supports the X.3 VT profile. From the VT client, issue a connection to the gateway, setting the first octet of the P selector to 4; the second octet may be any value. The X.25 address is part of the VT-profile parameters and is carried on the connect request, so no configuration on the gateway is required.

After configuring the X25 Service and Gateway Service parameters, see “Making Outgoing Connections” next for information on making outgoing automatic and extended connections.

Making Outgoing Connections

Before making automatic or extended outgoing connections, you must provide a list of connection addresses as well as connection and disconnection commands to end users of the connection service gateway. Configure IP connection addresses with the IPX25Map parameter or OSI connection addresses with the PSeIX25Map parameter.

The following is an example of IP-to-X.25 address mappings:

```
Config File      IP Address          X.25 Information
-----
! 2              192.9.209.100      R*311040800248
! 2              192.9.209.101      R,G09*311041502222
! 2              129.213.112.120    R-311041502222DHELLO
! 2              129.213.112.121    (PAD mode)
```

The following is an example of P-Selector-to-X.25 address mappings:

```
Config File      P-Selector          X.25 Information
-----
!2                01                  311040800248
!2                02                  (PAD Mode)
```

Users can make Telnet and VTP connections using the list of connection addresses. The connection command used depends on the commands that are available on the device from which the connection is made. For example, if a user initiates a connection request from a terminal connected to a 3Com communications server, then communications server commands such as the Connect command can be used. When connecting from another device, consult the documentation that ships with that device for information on commands that can be used, and make sure you provide the appropriate commands to users of the connection service gateway.

During outgoing connection establishment, the gateway selects a port through which the connection is made. These ports are not physical ports, but virtual ports. The gateway selects the next available port, and initializes the port and session with the specified configuration file (if none is specified, configuration file 2 is used), except OSI connections using X.3 profiles, which do not require the configuration file. No correlation exists between the selection of the port and the configuration file that initializes it. For example, the gateway could select port 8 and initialize it with configuration file 2. On the next connection to the same destination, the gateway could select port 60 and initialize with configuration file 2.

For information on making automatic connections, read the next section. For information on making extended connections, see "Extended Connections" later in this chapter.

Automatic Connections

When you initiate a connection request to the IP address of 192.9.209.101, for example, the gateway receives the request, locates the matching entry in the address mapping table, and uses the destination X.25 address and other information to place the call. In this example, the gateway expects to find a reverse charge facility offered at destination address 311040800248. The gateway also examines the following prefix table:

| Port #/IPAddr | X.25 Prefix |
|---------------|-------------|
| !4 | Default |
| !3 | 3110415 |
| !4 | 3110408 |

The gateway finds that prefix 3110408 can be reached on port 4. The gateway places the call to the destination X.25 host on port 4, and initializes the port and session with the parameter settings in configuration file 2. When the connection is established, the host prompt appears.

If the gateway cannot match the destination address with an address in the address mapping table and the prefix in the prefix table (no default path has been defined), the gateway rejects the connection and displays a message similar to "connection refused." The exact wording of the message is Telnet-client or VTP-client-dependent. If the gateway can match a destination address, but not a prefix, and a default port is defined for the X25Prefix parameter, the gateway uses the default port to place the call.

When you complete the session with the host, you need to end the session. The command that you use depends on the host.

Extended Connections

This section describes to an end user the packet assembler/disassembler (PAD) emulation mode features and how to make an extended connection. When you

initiate a connection request to the IP address of 129.213.112.121, for example, the gateway receives the request and locates the matching entry in the address mapping table that has no X.25 address. In this example, the gateway finds a destination match and places you into PAD mode, which is indicated by the NB-PAD> prompt.

When you establish an X.25 virtual call from PAD mode and the connection is established, the gateway displays on-screen messages indicating that you are connected. If the virtual call is rejected, the on-screen message is "CLR 0 0." For more information about establishing virtual calls, see "Establishing a Virtual Call" next.

In PAD emulation mode, you can perform the following actions:

- Select individual PAD parameter values.
- Request the current values of PAD parameters to be transmitted by the PAD to the host.
- Establish and clear a virtual call.

The PAD emulation user interface also supports the use of call user data and facilities with the command issued to establish a virtual call. Facilities include reverse charge requests and basic closed user groups.

The PAD emulation user interface provided by the X.25 connection service has limited functionality and only supports some of the capabilities described in CCITT Recommendation X.28. These supported capabilities and the command syntax for invoking them are described in the following sections.

Selecting Individual PAD Parameters

After selecting a default PAD profile, you can assign new values to individual parameters (overriding the default values) by using the SET command.

For example, to set the values of parameter 2 to 0, parameter 3 to 2, and parameter 9 to 4, at the NB-PAD> prompt, enter:

```
SET 2:0, 3:2, 9:4
```

To set the values of parameter 2 to 0, parameter 3 to 2, parameter 9 to 4, and to read the set values back, at the NB-PAD> prompt, enter:

```
SET? 2:0, 3:2, 9:4
```

Requesting Current Values of PAD Parameters

You can read the values currently assigned to individual PAD parameters by using the PAR? command. To read the current values for parameters 2, 3, and 9, at the NB-PAD> prompt, enter:

```
PAR? 2,3,9
```

Establishing a Virtual Call

You can establish a virtual call to an X.25 destination address by supplying the following information:

- The X.25 address

To establish a virtual call to a host whose X.25 address is 311040800248, enter the X.25 address at the NB-PAD> prompt as follows:

311040800248

- The X.25 address with optional call-user data

To establish a virtual call to a host whose X.25 address is 311040800248 and to transmit a call-user data string "HELLO" with the call request, enter one of the following strings at the NB-PAD> prompt:

```
311040800248DHELLO
311040800248PHELLO
```

The D and P distinguish the end of the address from the call-user data. Use D when you want the gateway to display the data string it is sending as call-user data on the call. Use P if you want to hide (protect) the data, for example, when sending passwords to the host.

- The X.25 address with optional facilities and with optional call-user data

The connection service supports two facility requests: reverse charge request and basic closed user group selection.

To establish a virtual call to a host whose X.25 address is 311040800248 and to request reverse charging, enter one of the following strings at the NB-PAD> prompt:

```
R-311040800248
R*311040800248
```

Either R- or R* can be used to indicate reverse charging.

To establish a virtual call to a host whose X.25 address is 311040800248 and to request a closed user group selection, enter one of the following strings at the NB-PAD> prompt:

```
G09-311040800248
G09*311040800248
```

Either G09- or G09* can be used to indicate closed user group.

To establish a virtual call to a host whose X.25 address is 311040800248 and to request both the reverse charging and closed user group, enter one of the following strings at the NB-PAD> prompt:

```
R,G09-311040800248
R,G09*311040800248
```

To establish a virtual call to a host whose X.25 address is 311040800248, to request both the reverse charging and closed user group, and to specify that the user data "HELLO" be transmitted as call user data with the call request, enter one of the following strings at the NB-PAD> prompt:

```
R,G09-311040800248DHELLO
R,G09*311040800248DHELLO
R,G09*311040800248PHELLO
```

- The path on the gateway to be used for establishing a connection

To establish a virtual call to a host that is connected over a private line and is not identified by an X.25 address, and to select the gateway path 3 (on which X.25 connection service is enabled) to be used for the connection, enter the following string at the NB-PAD> prompt:

```
L3
```

To specify that the user data "HELLO" be passed as call user data with the call request, enter one of the following strings at the NB-PAD> prompt:

L3DHELLO

L3PHELLO

To allow the gateway to select a line for establishing the call, enter the following string at the NB-PAD> prompt:

L

The gateway selects the lowest numbered path that is enabled for connection service.

When you establish a virtual call by using one of the previously described methods, the host displays a greeting or prompt, the appearance and format of which is host-dependent.

When you complete the session with the host, you need to end the session. The command used depends on the host. You can escape from the X.25 host by entering the PAD recall character (usually [Ctrl] + P) to return you to the NB-PAD> prompt, and return back to the X.25 host by entering another PAD recall character.

Clearing a Virtual Call

You can exit from the PAD mode prompt back to the original Telnet or VTP initiator by entering:

CLear

Troubleshooting Outgoing Connections

If you encounter problems with the connection service gateway, verify that the settings in the PATH, PORT, X25, LAPB (if used), and Gateway Services are correct as follows:

- To verify the control, state, baud, connector, and clock settings, enter:

SHow -PATH CONFiguration

- To verify that the owner of the wide area ports used in the connection service is X.25, enter:

SHow -PORT CONFiguration

- To verify the interface type, the X.25 address, and the PDN network type, enter:

SHow -X25 CONFiguration

- To verify the settings of the LAPB Service parameters, enter:

SHow -LAPB CONFiguration

For additional information, see the LAPB Service Parameters chapter in *Reference for Enterprise OS Software*.

- To verify the settings for the path used in the connection service, enter:

SHow -Gateway CONTrol

If connection requests continue to fail, enable the X.25 trace feature by using the SETDefault !<path> -X25 Trace = (Data, Control) syntax. Use SHow !<path> -X25 Trace to display data and/or control information for the specified X.25 path at the network layer. For more information about the Trace parameter, see the X25 Service Parameters chapter in *Reference for Enterprise OS Software*.



Make sure to turn Trace off after you are finished using it because it slows down the performance of your bridge/router.

You can enable tracing by using the SETDefault !<path> -Gateway CONTrol = Trace syntax to obtain additional information. After setting the CONTrol parameter, establish (or attempt to establish) a connection with the X.25 host. The screen displays information that can be used for debugging.

You can also display active session information such as the source and destination address by entering:

SHow -Gateway PadSession.

For more information on the CONTrol and PadSession parameters, see the Gateway Service Parameters chapter in *Reference for Enterprise OS Software*.

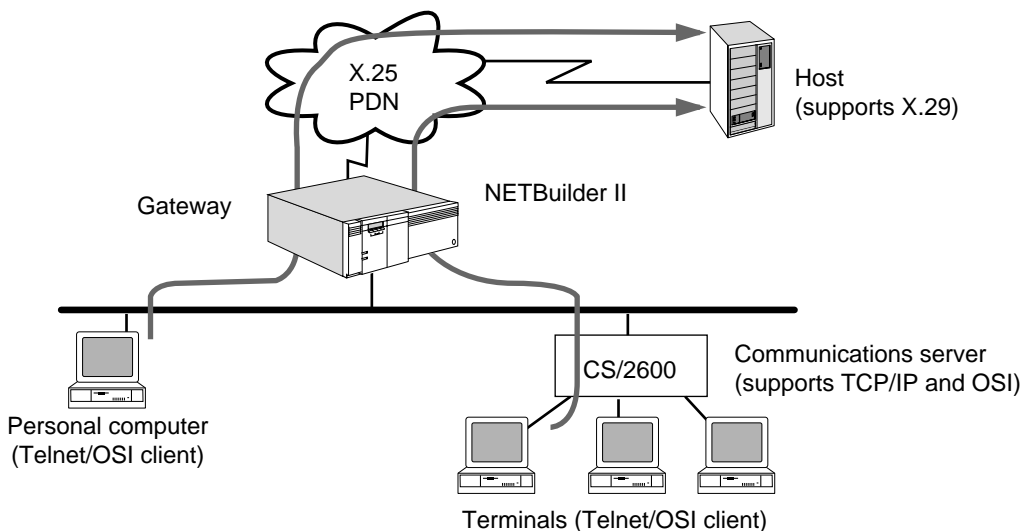
How the Outgoing Connection Service Works

The X.25 connection service gateway allows IP Internet-attached Telnet clients and OSI VTP clients to connect to X.25-attached hosts that support the X.29 Protocol. The Telnet or OSI VTP clients can be PCs or workstations running Telnet client software or VTP client software, or asynchronous dumb terminals connected to a communications server that supports the Telnet and/or VTP protocol. LAN-to-WAN connections are also referred to as *outgoing connections* and are controlled by the outgoing connection service of the gateway. Figure 400 is an example of outgoing LAN-to-WAN connections.

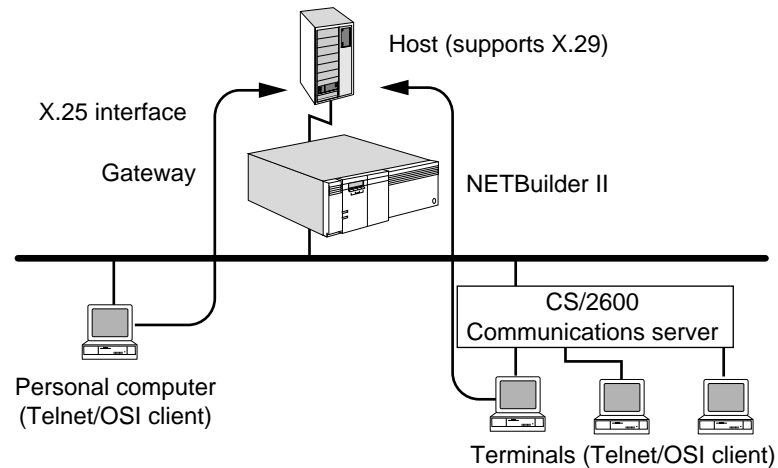


Login is not supported on outgoing calls.

Figure 400 LAN-to-WAN Connections (Outgoing)



The X.25 connection service can also be used to front-end a host that does not support a LAN interface, but has an X.25 interface and supports the X.29 Protocol as shown in Figure 401. This configuration is very similar to the one shown in Figure 400, except that neither the connection service gateway nor the host is connected to an X.25 public data network (PDN). The gateway and the host instead are connected directly to each other back-to-back, using X.25 for terminal connections.

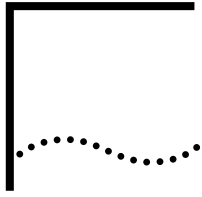
Figure 401 Host Front-End Connections (Outgoing)

The X.25 connection service gateway offers two types of outgoing connections:

- Automatic (one-step)
End users can enter a connection command from the Telnet client or OSI VTP client and the gateway automatically establishes the link to the X.25 host.
- Extended (two-step)
End users can enter a connection command from the Telnet client or OSI VTP client and establish a connection with the gateway's PAD emulation user interface. Once in PAD emulation mode, users can execute a connection command to the desired host by providing the appropriate information.



With outgoing connections, you are limited to connecting to a single host with each Telnet or VTP connection.



CONFIGURING CONNECTIONS FOR INCOMING CALLS

This chapter describes how to configure your bridge/router to function as an X.25 connection service gateway for incoming calls. The gateway allows end users to make connections from X.25 packet assembler/disassembler (PAD)-attached terminals to IP Internet-attached Telnet, Rlogin servers, or Rlogin hosts. This chapter describes procedures for making incoming automatic (one-step) and extended (two-step) connections, configuring name services for Transmission Control Protocol/Internet Protocol (TCP/IP) connections, configuring Rlogin connections, and selecting the name service for Open System Interconnection (OSI) connections.



For conceptual information, see “How the Incoming Connection Service Works” later in this chapter.

Configuring the Gateway for Incoming Connections

This section describes how to configure the bridge/router gateway to handle incoming connections.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the system's local and wide area ports and paths by referring to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.
- Configure the X.25 interface.

After completing the procedure for configuring local area and wide area paths and ports, configure X.25 as the owner of each wide area interface to be used in the incoming connection service using:

```
SETDefault !<port> -PORT OWNEr = X25
```

To configure the X.25 interface, see the X25 Service Parameters chapter. You may also want to use the data compression feature; for detailed information, see the Configuring Data Compression chapter.

You must configure the gateway paths for incoming automatic and extended connections before you configure the bridge/router. See “Making Incoming Connections” later in this chapter.

Procedure To configure the gateway paths for incoming automatic and extended connections, follow these steps:

- 1 Verify that the gateway paths are configured for incoming automatic and extended connections.

For example, to display information for path 3, enter:

```
SHow !3 -Gateway CONTrol
```

The following display appears:

```
Path !3 CONTrol = (Enable, InExt, OutExt, InAuto, OutAuto, DDXP, SubAddr,
NoDSA, NoTrace)
```

By default, the CONTrol parameter is disabled. However, it automatically becomes enabled when the X.25 path comes up.

- 2 If the display is incorrect, configure the path for the desired type of operation using:

```
SETDefault !<path> -Gateway CONTrol = (Enable, InAuto, InExt)
```

After configuring the X25 Service and the Gateway Service parameters, see the next section for information on making incoming automatic and extended connections.

Making Incoming Connections

To initiate a connection request from an X.25 PAD-attached terminal, end users must use commands that are supported by the PAD service provider. Examples provided in this section use a general command syntax and consist of elements that are generally made available by PAD service providers.

During incoming connection establishment, the gateway selects a port through which the connection is made. These ports are not physical ports, but virtual ports, and range in number from 0 to 127 on the NETBuilder II system. The gateway selects the next available port, and initializes the port and session with the specified configuration file (if none is specified, configuration file 1 is used). No correlation exists between the selection of the port and the configuration file that initializes it. For example, the gateway could select port 3 and initialize it with configuration file 1. On the next connection establishment to the same destination, the gateway could select port 7 and initialize it with configuration file 1. For information on configuration files, see "Using Configuration Files" and "Creating Port-Initialization Macros" later in this chapter.

For information on making incoming automatic connections, see the next section. For information on making incoming extended connections, see "Extended Connections" later in this chapter.

Automatic Connections

When making an automatic connection request from the PAD-attached terminal to a Telnet, Rlogin, or OSI server, you must identify the X.25 address of the gateway in addition to the destination server. You can specify the destination server as X.25 Call User Data (data to be sent to the gateway along with the call request) in one of the following ways:

- Host address (IP address)
- Host name (IP or OSI)
- Configuration file number

Automatic incoming connections are also supported for subaddress mapping. For more information on configuring a subaddress map, see the description for the SubAddrMap parameter in the Gateway Service Parameters chapter in *Reference for Enterprise OS Software*.

Using Addresses

To connect a PAD-attached terminal user to the Telnet server whose IP address is 129.213.112.009:

```
<connect> 311040800245 D 129213112009
```



You must substitute a connection command for <connect> that is supported by the PAD service provider.

This command specifies the gateway's X.25 address of 311040800245; X.25 Call User Data follows and specifies the destination IP address of 129.213.112.009. When you supply the IP address as Call User Data, be sure to include zeros. For example, do not write 129213112009 as 1292131129. The letter D separates the X.25 address from the Call User Data.

The gateway uses the addressing information to automatically place the call to the destination. The gateway selects the next available port, and initializes the port and session with configuration file 1. To disconnect the session, use an exit command. The specific command that is entered is host-dependent.

Using Names

To connect a PAD-attached terminal user to the gateway's X.25 address and to a server named "marketing" use:

```
<connect> 311040800245 D marketing
```



You must substitute a connection command for <connect> that is supported by the PAD service provider.

When you use a name to make an incoming automatic connection, the name can be no longer than 12 characters to conform to X.25 Call User Data limitations.

This command specifies the gateway's X.25 address of 311040800245; Call User Data follows with a name. The name "marketing" identifies the host; the gateway resolves the name through the IP (Domain name or IEN116) name resolver, or the OSI Name Server (X.500 or File) and automatically places the call to the destination. The gateway selects the next available port, and initializes the port and session with configuration file 1. To disconnect from the session, use an exit command. The specific command that is entered is host-dependent. For network manager information on configuring name services, see "Name Service for TCP/IP Connections" or "Name Service for OSI Connections" later in this chapter.

Using Configuration Files

To automatically connect a PAD-attached terminal user to the gateway's X.25 address and to a server whose address or name is specified in a port-initialization macro called by a configuration file use:

```
<connect> 311040800245 D 03
```



You must substitute a connection command for <connect> that is supported by the PAD service provider.

This command connects a PAD-attached terminal user to the gateway's X.25 address of 311040800245; Call User Data follows with a configuration file number. The configuration file calls a port-initialization macro, which contains a TELnet, RLOGin, or VTp connection command to a server.

During connection establishment, the gateway selects the next available port, initializes the port and session with the settings in the specified configuration file, and automatically places the call to the destination.

The gateway supports only one session with incoming automatic connections. Access control must be disabled for automatic connections using the configuration file.

To disconnect the session, use an exit command (host-dependent).

For network manager information about creating port-initialization macros, see "Creating Port-Initialization Macros" later in this chapter.

Extended Connections

An extended connection request occurs in two steps. First, you must identify the X.25 address of the gateway in the connection command and make a connection to the gateway. Second, you can connect to Telnet or Rlogin servers on the IP Internet, or to an OSI host. You also can configure and manage the 3Com bridge/router.

To make an extended connection, follow these steps:

- 1 Make an X.25 connection to the gateway using:

```
<connect> 311040800245
```



You must substitute a connection command for <connect> that is supported by the PAD service provider.

If access control is disabled, you are placed into the management/configuration interface, indicated by the Enterprise OS> prompt. The interface is the same interface that is seen when you connect to the system through a local console, through a Telnet connection, or through an OSI VTP connection.

If access control is enabled, you must supply a valid user name and password assigned to you by the network manager at the Netlogin prompt. If the name is invalid, the gateway rejects the connection; otherwise, you are placed into the bridge/router's user interface. For network manager information on configuring access control, see the Configuring Local Access Control chapter.

When you have the Enterprise OS> prompt, you can make connections to Telnet, Rlogin, or VTP hosts as described in step 2. If you have Network Manager privilege and want to manage or configure the bridge/router, proceed to step 3.

- 2 From the Enterprise OS> prompt, make connections to TCP/IP or OSI hosts by using the gateway connection service commands.

You can use the TELnet, RLOGin, or Connect commands to connect to a TCP/IP host. With each of the connection commands, you can use a name or an IP address. For example, you can enter one of the following commands:

Telnet 129.213.112.9
Connect host1

You can use the VTp or Connect commands to connect to an OSI host. With these commands, you can use a name or an OSI address. For example, you can enter one of the following commands:

VTp /47/0004/003511003C3C3C5C3C3C01 !9
Connect /47/0004/003511003C3C3C5C3C3C01 !9

N-selectors, T-selectors, and S-selectors in the PSAP address are host-dependent. For more information about PSAP addressing, see the NSAP and PSAP Addressing appendix.

If you use a name, the gateway performs the name resolution through the IP (Domain name or IEN116) name resolver, or the X.500 DUA. For network manager information on configuring name services, see “Name Service for TCP/IP Connections” or “Name Service for OSI Connections” later in this chapter.

You can also make connections to multiple destinations by entering ECM with the Connect, TELnet, RLOGin, and VTp commands. For more information, see the Managing Sessions for Incoming Extended Calls chapter.

- 3 From the Enterprise OS> prompt, manage or configure the bridge/router.

You must have Network Manager privilege and not be restricted through access control. For network manager information on configuring access control, see the Configuring Local Access Control chapter.

- 4 When you complete the extended session with the host, end the session.

The commands used to exit or logout are provided by the host. After entering an exit command, you are returned to the Enterprise OS prompt.

- 5 To disconnect your session from the gateway and return to the PAD terminal prompt, enter one of the following commands:

Listen
LOGout

If you have Network Manager privilege, you can disconnect another user’s session by specifying their port number. The following command disconnects the user’s session on port 3 and puts the port into listen mode:

LOGout !3

For additional information about these commands, see the Commands chapter in *Reference for Enterprise OS Software*.

Troubleshooting Incoming Connections

If you encounter problems with the connection service gateway, verify that the settings in the PATH, PORT, X25, LAPB (if used), and Gateway Services are correct as follows:

- To verify the control, state, baud, connector, and clock settings, enter:
SHow -PATH CONFIguration
- To verify that the owner of the wide area ports used in the connection service is X.25, enter:
SHow -PORT CONFIguration
- To verify the interface type, the X.25 address, and the public data network (PDN) network type, enter:

SHow -X25 CONFIguration

- To verify the settings of the LAPB Service parameters, enter:

SHow -LAPB CONFIguration

- For additional information, see the LAPB Service Parameters chapter in *Reference for Enterprise OS Software*.
- To verify the settings for the path used in the connection service, enter:

SHow -Gateway CONTrol

If connection requests continue to fail, enable the X.25 trace feature using:

```
SETDefault !<path> -X25 Trace = (Data, Control)
```

To display data and control information for the specified X.25 path at the network layer, use the SHow !<path> -X25 Trace syntax. For more information about the Trace parameter, see the XSWitch Service Parameters chapter in *Reference for Enterprise OS Software*.

Enable tracing using the SETDefault !<path> -Gateway CONTROL = Trace syntax to obtain additional information. After setting the CONTROL parameter, establish (or attempt to establish) a connection with the X.25 host. The screen displays information that can be used for debugging.

To display a history of the status of the last few sessions, use the SHow !<path> -Gateway ConnHistory syntax. Additional information on the CONTROL and ConnHistory parameters can be found in the Gateway Service Parameters chapter in *Reference for Enterprise OS Software*.

Customizing the Incoming Connection Service

This section describes how you can customize the incoming connection service.

Creating Port-Initialization Macros

You can specify a configuration file as Call User Data to be used with an incoming automatic connection request (for example, connect 311040800245 D 01). After receiving the connection request, the gateway uses the configuration file to call a port-initialization macro previously assigned to a configuration file by the network manager through the -TERM InitMacro parameter. The macro must contain a connection command to a Telnet, Rlogin, or OSI host to automate the incoming connection. During connection establishment, the gateway selects the next available port, initializes the port and session with the contents of the configuration file, and makes the connection to the host through the macro.

Although configuration files numbered 1 through 32 are valid for incoming automatic connections using configuration files, you should use configuration file 1 as the default. You can usually use configuration file 1 without modification; the default settings of the TERM Service parameters are acceptable for most incoming connections. If you require different settings from the defaults already provided, use one of the configuration files numbered 3 through 32.



Configuration file 2 is the default for outgoing connections and must not be used for incoming connections. If you use an even-numbered configuration file for an incoming connection, make sure you change the DeVice parameter from Host to

Terminal using the SETDefault !configfile -TERM DeVice = Terminal command, or the connection attempt will fail.

This section describes how to:

- Create and manage macros.
- Assign a macro to a configuration file number.

Table 94 lists commands that are used for creating and managing macros, and assigning the macro to a configuration file number. You can find more detailed macro information in the Macro Features appendix.

Table 94 Commands for Creating and Managing Macros

| Procedure | Command | Function |
|--|--|--|
| Creating macros | DEFine <macroname> = <text> | Defines a macro. |
| | PAuse | Makes the system pause for one second. |
| | PAuse <seconds> | Makes the system pause for the specified number of seconds during macro execution. |
| | Echo "<string>" | Displays the specified string on the terminal during macro execution. |
| Assigning a macro to a configuration file number | SETDefault !<configfile> -TERM InitMacro = "<macroname>" | Defines port-initialization macros. |
| Managing macros | SETDefault !<configfile> -TERM InterActTerm = NoMacroEcho | Suppresses the display as the macro is executed. |
| | SETDefault !<configfile> -TERM InterActTerm = NoMacroBreak | Prevents macro termination with the Break key. |
| | Press Break key | Stops execution of the macro. |
| | UNDefine <macroname> | Deletes the macro. |
| | FLush -SYS MACros | Removes all macros from the macro cache. |
| | SHow -SYS MACros | Displays all macro names. |
| | SHow -SYS MACros <macroname> | Displays contents of specified macros. |

Creating Macros

A macro is a file that contains a series of individual commands that automates the incoming automatic connection. You can create a macro if you have Network Manager privilege. The macro can consist of a connection command or a series of connection commands in a menu-driven interface, as well as X.3-type parameter settings in the TERM Service that are used to initialize the session with the host.

After the macro is created, you can assign the macro as a port-initialization macro and give it a configuration file number. Each time the X.25 PAD-attached terminal user specifies the configuration file number as Call User Data in the incoming automatic connection request, the gateway automatically executes the port-initialization macro that contains the connection command.

You use the DEFine command to create a macro file and specify its contents. When a new macro is created with the same name as an existing macro, the new macro contents replace the old macro contents.

A single macro can contain up to 256 characters. Macro contents must begin with a left parenthesis. If the definition requires more than one line, press the Return key after the opening parenthesis. After you press the Return key, the Macro: prompt appears on the next line. All characters entered between the opening and closing parentheses are part of the macro. Nested parentheses in balanced pairs are allowed.

To create a macro, follow these steps:

- 1 Create the macro using:

```
DEfine <macroname> = (
```

Macro names can be up to 14 characters long; the first character must be alphabetic. Names longer than 14 characters are truncated. The macro service does not distinguish between upper- and lowercase letters in macro names.

For example:

```
DEfine start = (
```

The name of this macro is "start", the left parenthesis indicates the beginning of the macro.

After pressing the Return key, the Macro: prompt appears on the next line.

- 2 Enter the desired commands at the Macro: prompt.

For example, the following commands request a connection to "host1," pause for a second to give time for the host to respond with a login request, transmit the user's name as the login name, and transmit the password:

```
Echo "connection"
Connect host1 ECM
PAuse 1
TRansmit "terry"
PAuse 1
TRansmit "<password>"
RESume
```

The text of the macro must conform to the conventions for assigning strings described in *New Installation for NETBuilder II Software*.

The Break key or the character specified by the -TERM BReakChar parameter can be used to cancel the DEfine command at any time before the terminating right parenthesis is entered.

- 3 Complete the macro by entering the right parenthesis.

The normal Enterprise OS command prompt returns.

- 4 Suppress the display of the macro as the macro is executed, and make sure the user can terminate the macro execution with the Break key using:

```
SETDefault !<config file> -TERM InterActTerm = (NoMacroEcho, MacroBreak)
```

NoMacroEcho and MacroBreak are the default settings of the -TERM InterActTerm parameter and may not need to be set.

Assigning the Macro to a Configuration File

After defining the macro, you need to assign it as a port-initialization macro and give it a configuration file number. To assign the macro to a configuration file, use:

```
SETDefault !<config file> -TERM InitMacro = "start"
```

Valid configuration files numbers are 1 and 3 through 32. Configuration file 2 is the default for outgoing connections and must not be used for incoming connections.

This command assigns the macro named "start" as a port-initialization macro to a configuration file. When the PAD-attached terminal user initiates an incoming automatic connection using the specified configuration file, the gateway selects the next available port, initializes the port and session with the contents of configuration file, changes the port from listen mode to command mode, executes the initialization macro, and makes the connection request to host1.

Managing Macros

Use the following commands to delete and display macros, and to flush the macro cache:

- To delete a macro, use:

```
UNDefine <macroname>
```

In this command, <macroname> is the name of the macro to be deleted.

- To display all the defined macros on the gateway, enter:

```
SHow -SYS MACros
```

- To display the contents of a specific macro, use:

```
SHow -SYS MACros <macroname>
```

- To remove all macros from the macro cache, enter:

```
FLush -SYS MACros
```

Name Service for TCP/IP Connections

Because users more easily remember names instead of addresses, the X.25 connection service software allows you to assign names to IP addresses. Using names also helps users associate a network resource with its function and allows users to connect to resources without knowing their network addresses.

Name and corresponding address information is maintained in a database provided by the name service. This service maintains and updates information regarding resource names and addresses, and responds to queries regarding names. The network manager decides which one of the following two name services to use:

- IEN116
 - Allows you to use a database maintained on the gateway disk to add, remove, and change the names of network resources. The IEN116 name service can be stored on any gateway or terminal server with an internal diskette.
- Domain
 - Allows the gateway to use, but not provide, the Domain name service. The network must include a Domain name server that responds to Domain name requests from the gateway. The Domain name service is more widely used than IEN116.

If your gateway is installed on a network that is already in operation, the name service probably has already been defined. Set up your gateway to use the existing name service. If you are setting up a new network, you need to select the name service for it.

If you plan on using names in incoming automatic or extended connection requests to TCP/IP hosts, you need to:

- Select either the IEN116 or the Domain name service.
- Select the address of the primary and/or secondary name servers.
- Assign names to network addresses or resources.

To configure the gateway for Domain name service, see “Domain Name Service” earlier in this chapter.

Domain Name Service

Because the gateway does not implement the server side of the Domain name service (but does implement the client side), your network must include a Domain name server. The Domain name server responds to Domain name requests from the gateway. All names must be added and removed from the Domain name database on the Domain name server.

If you select Domain name service, you must specify a primary name server if your gateway boots from an internal diskette. You can optionally specify a secondary name server to be used when the primary name server is unavailable.

Configuring the Gateway To configure the gateway to use the Domain name service, follow these steps:

- 1 Specify the string for the DomainName parameter using:

```
SETDefault -IPName DomainName = "<name>"
```

The domain name can be no more than 128 characters. The domain name resolver appends the domain string to each name request that does not include a period.

- 2 Specify the primary name server address using:

```
SETDefault -IPName PrimaryNameServer = <address>
```

You can specify the secondary name server address using:

```
SETDefault -IPName SecondaryNameServer = <address>
```

- 3 Assign names to the resources on your network from the name server.

Consult the name server documentation for details on how to do this.

Using the Name Cache The gateway uses a storage area in memory, called the *name cache*, where some previously used names and their corresponding addresses are stored. The name cache allows the gateway to recall domain names quickly and reduces the data traffic load on the network. When a domain name request is made, the gateway searches its name cache first. If the information is not found, the resolver refers the name request to the primary name server.

If the Network Manager modifies any of the domain names or the information database is otherwise corrupted, discrepancies can exist between the information stored in the local cache and in the Domain name server. To correct these discrepancies by deleting cache information, enter:

```
FLush -IPName CAChe
```

This command deletes all entries in the cache. In addition, the cache is cleared automatically whenever the DomainName parameter is changed.

To display the contents of the name cache, enter:

```
SHow -IPName CAChe
```

Configuring Rlogin Connections

The RLOGin command can be used in incoming extended TCP connections to a specified IP Internet-attached host that is using the Rlogin protocol. Because Rlogin supports passing on additional information during session establishment, you may need to do more configurations.

The 3Com implementation of Rlogin supports the Rlogin client only. The client username (username on the client side), server username (username to be used for login on the Rlogin server side), terminal type, and baud rate are communicated to the server during the connection setup. The number of rows and columns also can be communicated to the server if the server requests the information.

To configure for Rlogin connections, obtain Network Manager privilege follow these steps. Depending on your configuration, some changes may be optional.

- 1 Set the terminal type to be used for the Rlogin connection using:

```
SETDefault !<configfile> -TERM TERMTType = "<string>"
```



Use configuration file 1 for incoming connections. If you need to customize a configuration file for a specific host, use configuration files 3 through 32. You must not use configuration file 2 because it is the default for outgoing connections. If you use an even-numbered configuration file for an incoming connection, make sure that you change the DeVice parameter from Host to Terminal using the SETDefault !configfile -TERM DeVice = Terminal command or the connection attempt will fail.

The gateway transmits the string to the Rlogin server when the RLOGin command is used. The TELnet command also uses this string, which has a maximum of 40 characters.

- 2 To set the number of columns and rows for the terminal, use:

```
SETDefault !<configfile> -TERM COLumns = <number> (1-255)
```

```
SETDefault !<configfile> -TERM ROW = <number> (1-255)
```

- 3 To specify that the gateway send an empty string for the client username to the destination during the connection negotiation, enter:

```
SETDefault -TCPAPPL RLogSendName = No
```

With this setting, the user is usually prompted for a password to log on to the remote host.

The default for this value is "yes." If this value is used, the client username is automatically sent to the destination host.

Name Service for OSI Connections

Using names in connection requests helps users to associate a network resource with its function. Also, connections to resources can occur without users knowing the network addresses of the resource. Because incoming automatic connections to OSI hosts are identified by names and not addresses, you must configure the gateway to assign names to presentation service access point (PSAP) addresses to use OSI name services. However, extended connections and incoming automatic connections (which use the configuration file) can use addresses.

Name and corresponding address information is maintained in a database provided by the name service or directory service. The name service maintains and updates information regarding resource names and their corresponding addresses and responds to queries regarding names. The gateway supports the following two types of name services:

- X.500 directory service

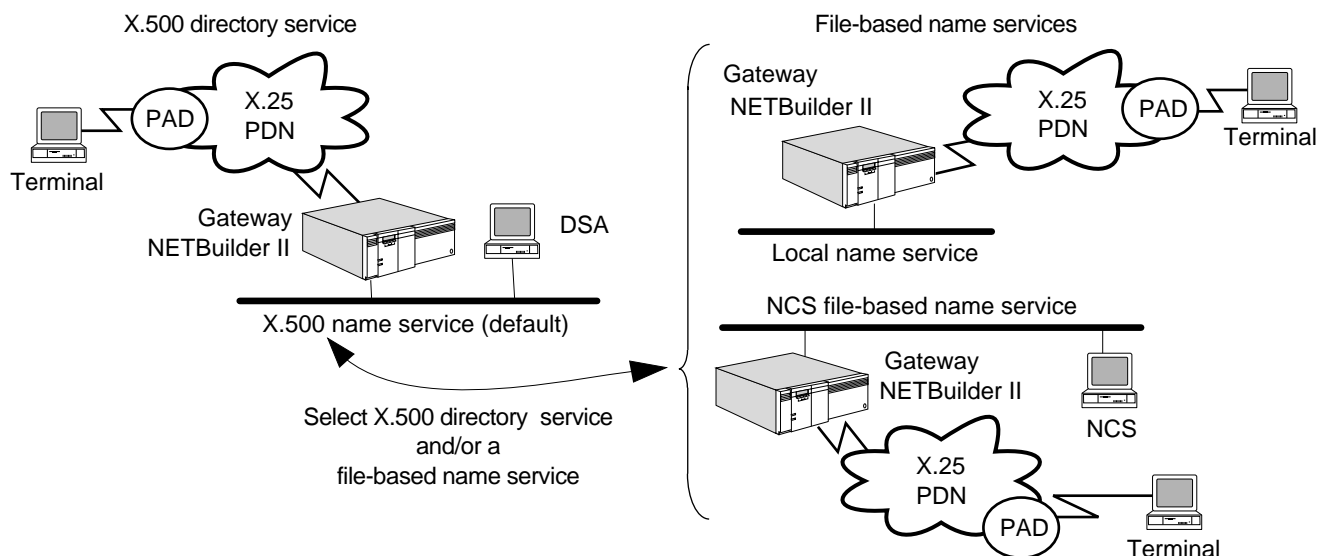
Allows you to use a database called the Directory Information Tree (DIT) which is maintained on a computer that runs the X.500 protocol. With this database, you can add, remove, and show the names and addresses of network resources from the gateway.

- File-based name service

Stores the database on the gateway diskette. By default, a gateway that boots from its own diskette stores the database on the gateway diskette.

Figure 402 shows these two name services.

Figure 402 Name Services Supported for Incoming OSI Connections



You can use one or both name services. If you use both name services, one name service first attempts to resolve a name request from the gateway and if that fails, the other name service attempts to resolve it. You can configure the order of this operation by entering:

```
SETDefault -OSIAPPL NameSourceOrder
```

When selecting the name service, remember that if you use a name service that has its database stored on another computer, the same database can be used for multiple servers. If you store the name service on the local diskette, a separate name service must be set up for each server.

If you have a computer that can support X.500 directory service, use the X.500 directory service.

To configure name services for incoming OSI connections, you need to:

- Select either the X.500 or the file-based name service, or both.

- Determine the name resolution order, if both name services are used.
- Assign names to resources.

To configure the gateway to use the X.500 directory service, see the next section. To configure the gateway to use the file-based name service, see “File-Based Name Service” later in this chapter.

X.500 Directory Service

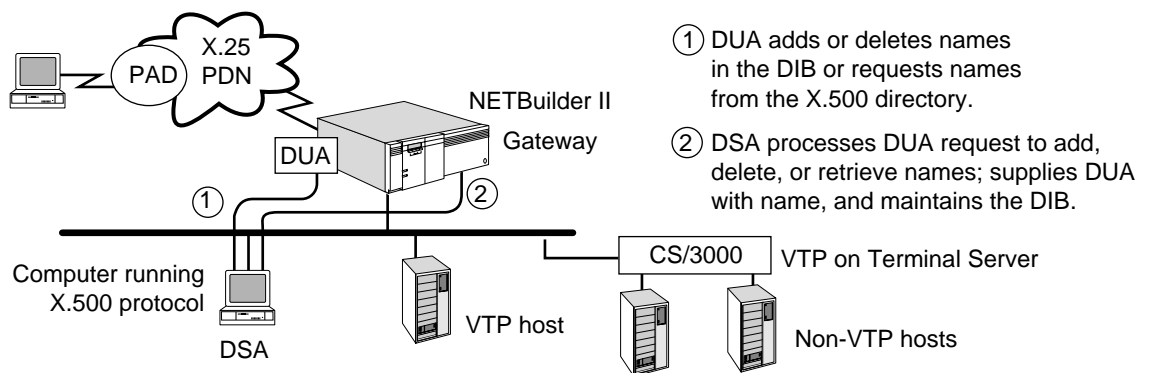


If you do not have a computer on your network that supports the X.500 protocol, skip this section and continue to “File-Based Name Service” later in this chapter.

The X.500 directory service allows you to use a database called the Directory Information Tree (DIT), which is maintained on a computer that runs the X.500 protocol. With this database, you can add, remove, and show the names and addresses of network resources from the gateway. Additional information about the DIT is provided in “Adding Entries” later in this chapter.

The computer running X.500 contains the Directory System Agent (DSA) and the Directory Information Base (DIB) database. The DSA maintains the DIB and interfaces with the Directory User Agent (DUA) that runs on the gateway. The DSA processes the DUA operation requests, such as “add a name” or “delete a name” as shown in Figure 403.

Figure 403 DUA and DSA Interaction



The DUA sends a request to the DSA when it needs the DSA to resolve a name to a presentation address during a VTP <name> or Connect <name> command (available in incoming extended connections), or when you want to access the DIB using the DirectoryManage command.

Configuring the Gateway To configure the gateway for X.500 directory service, follow these steps:

- 1 Confirm that the NameSourceOrder parameter includes X.500 in its values by entering:

```
SHow -OSIAPPL NameSourceOrder
```

If you want to use both name services, include both names and the order in which name requests should be resolved.

For example, to include both X.500 and file-based name services and to specify that the X.500 directory be queried first, enter:

```
SETDefault -OSIAPPL NameSourceOrder = X500 File
```

If you want to use only the X.500 directory service, enter:

```
SETDefault -OSIAPPL NameSourceOrder = X500
```

2 Specify the address of the DSA using:

```
SETDefault -OSIAPPL DSAddress = <PSAP address>
```

You may need to change the address of the DSA first. To accomplish this, disconnect the gateway from its current DSA by entering:

```
UnBindDSA
```

Set the address of the new DSA using:

```
SETDefault -OSIAPPL DSAddress = <PSAP address>
```

The DUA-DSA connection is transparent to the user. The connection occurs by either an operation request to the DSA or an incoming extended connection attempt made with the VTp <name> or the Connect <name> command.

3 Select a DSA vendor by entering:

```
SETDefault -OSIAPPL DSAType = Standard
```

4 Use the DirectoryManage command to add names of resources on your network.

The DirectoryManage command uses a menu system to add directory names. For information on using DirectoryManage and completing the configuration procedure, see the next section.

Managing Entries in the DIB The menu-driven DirectoryManage command allows you to add, remove, and show entries in the DIB. Each entry in the DIB is made up of attributes. These attributes depend on the object class the entry describes. Examples of object classes are "Country" or "Person." Attributes of the object class "Person" could be "Name," "Social Security Number," and "Address." For example, a typical entry belonging to the object class "Person" could be:

```
{Name = John Doe, SS# = 543-45-4333, Address = 324 Bayfront Ave., Santa Clara}
```

The gateway supports the following object classes: Country, Organization, OrganizationalUnit, ApplicationProcess, and ApplicationEntity. Their respective attributes are CountryName, OrganizationName, OrganizationUnitName, CommonName, and PresentationAddress.

Entries in the DIB are arranged in the DIT. Figure 404 and Figure 405 show the tree structure and how it applies to a directory name. The position of the object classes in the tree reflects their hierarchical relationship. Country is highest in the tree, followed by Organization, OrganizationalUnit (up to 3 levels are allowed), ApplicationProcess, and ApplicationEntity. This hierarchy must always be respected when configuring a DUA operation.

A leaf entry, which is an entry without any entries below it, is the only type of entry that can be added or deleted. For example, in the entry {CountryName US, OrganizationName 3Com}, US is not a leaf entry because 3Com is below it; therefore, it cannot be deleted. 3Com is a leaf entry and can be deleted.

Figure 404 Directory Information Tree (DIT)

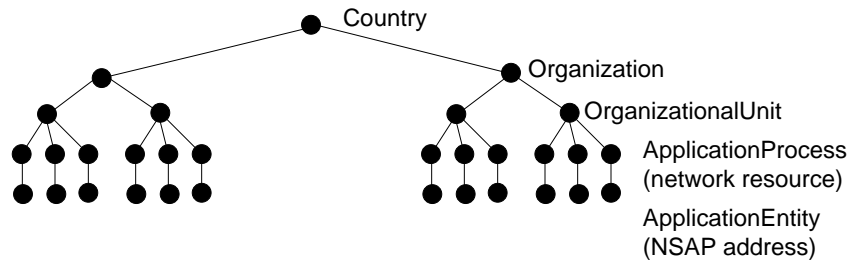
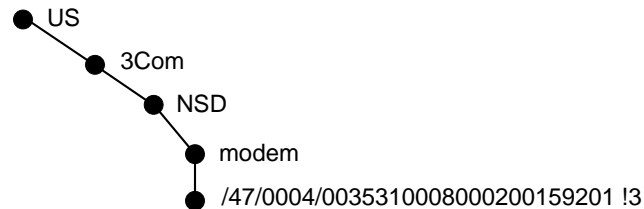


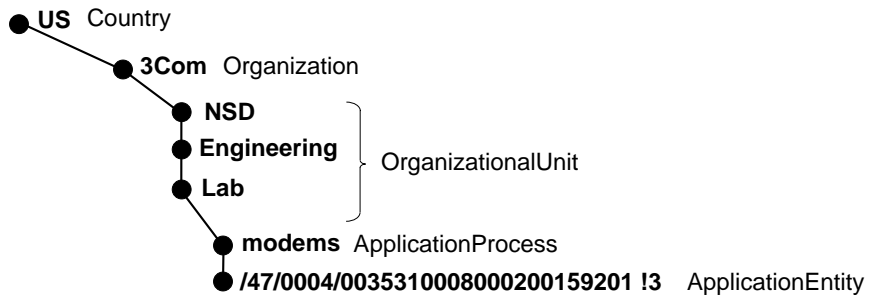
Figure 405 Directory Name in DIT Format



The object classes are defined as follows:

- **Country**
Common to all directory names in the same directory. In the example in Figure 405, country is defined as US, so this is the first part of all names in this directory. The country name must consist of two characters.
- **Organization**
A maximum of 14 characters is allowed. In the example, 3Com is the organization name.
- **OrganizationalUnit**
Up to three levels of organizational units can exist in the DIB. In the example, NSD is an organizational unit. The OrganizationalUnit name cannot be more than 14 characters. Figure 406 shows a directory with three levels of organizational units.

Figure 406 Directory Name with Three Organizational Units



- **ApplicationProcess**
CommonName is the attribute of the ApplicationProcess object class; it refers to the name of the network resource. In Figure 406, modems is an application process.

- ApplicationEntity

The address that corresponds to the resource name, or presentation address, is the attribute of the ApplicationEntity object class. See Figure 406 for an example of an application entity.

Adding Entries To add an entry to the X.500 directory, select the Add Name option from the Directory Manager Menu, and then enter the directory name.

To assign the name "C=US O=3Com OU=NSD CN=modems" to the PSAP address of the gateway port to which modems are attached, follow these steps. It assumes that no country name has been defined in the directory name database.

- 1 Specify US as the country name by doing the following:

- a To invoke the Directory Manager menu, enter:

DirectoryManage

The following main menu is displayed:

```
-----Directory Manager Menu-----
1.- Add name
2.- Delete name
3.- List allnames
4.- Print one VT name
5.- Set user name and password
6.- Set default DN
```

- b Select option 1, Add name.

The following submenu is displayed:

```
-----Directory Manager Menu-----
No Default Distinguished Name
No UserName
1.- Country
2.- Organization
3.- OrganizationalUnit
4.- ApplicationProcess
5.- ApplicationEntity
6.- Do Add Request
```

Remember to follow the hierarchy and only add leaf entries. You must add one level at a time, starting from the top. For example, to add the entry {CountryName US, OrganizationName 3Com, OrganizationalUnitName Engineering}, you must first add the entry {CountryName US} to the DIB. You then add the entry {CountryName US, OrganizationName 3Com}, and finally the entry {CountryName US, OrganizationName 3Com, OrganizationalUnitName Engineering}.

- c Select 1 to specify the country name.

- d For the country name, enter:

us

- e Select 6, Do Add Request, to add the country name.

If a country name has already been specified in the database, a message appears.

- 2 Specify the organization name, 3Com, by following these steps:

- a Select 1 from the Directory Manager submenu and when prompted for the country name, enter:
us
 - b Select 2 from the Directory Manager submenu.
 - c For the organization, enter:
3Com
Up to 14 characters can be entered for the organization name.
 - d Select 6, Do Add Request, to add the name C=US O=3Com.
 - 3 Specify the organizational unit, NSD, by following these steps:
 - a Select 1 from the Directory Manager submenu and enter **us** when prompted for the country name.
 - b Select 2 from the Directory Manager submenu and when prompted for the organization name, enter:
3Com
 - c Select 3 from the Directory Manager menu.
 - d For the organizational unit name, enter:
NSD
 - e Select 6, Do Add Request, to add the name C=US O=3Com OU = NSD.
 - 4 Specify the Application Process, modems, by following these steps:
 - a Select 1 from the Directory Manager submenu and when prompted for the country name, enter:
us
 - b Select 2 from the Directory Manager submenu and when prompted for the organization, enter:
3Com
 - c Select 3 from the Directory Manager submenu and when prompted for organizational unit, enter:
NSD
 - d Select 4 from the Directory Manager submenu.
 - e At the CommonName prompt, enter:
modems
A maximum of 14 characters can be specified for the CommonName attribute.
 - f Select 6, Do Add Request, to add the name C=US O=3Com OU=NSD CN=modems.
 - 5 Specify the Application Entity by following these steps:
 - a Select 1 from the Directory Manager submenu and when prompted for the country name, enter:
us
 - b Select 2 from the Directory Manager submenu and when prompted for the organization name, enter:
3Com

- c Select 3 from the Directory Manager submenu and when prompted for the organizational unit name, enter:
NSD
- d Select 4 from the Directory Manager submenu and enter when prompted for the CommonName of the application process.
modems
- e Select 5 from the Directory Manager submenu.
- f Enter the PSAP address for modems that is connected to gateway 2:
/47/0004/0035310008000200159201
- g Select option 6, Do Add Request, to add the name C=US O=3Com OU =NSD CN=modems PA=/47/0004/0035310008000200159201.

Displaying Directory Names Use the DirectoryManage command to either display all names or display the address of a particular name.

To display all names, follow these steps:

- 1 Select option 3, List all names, from the Directory Manager main menu.
- 2 Specify Country, Organization, OrganizationUnit and Filtered Application Process, if necessary.

If you are using a Default Distinguished Name, see "Setting Up the Default Distinguished Name" later in this chapter.

- 3 Select option 5, Do list request.

To display the address of a particular name, follow these steps:

- 1 Select option 4, Print one VT name.
- 2 Specify the parts of the name.
- 3 Select option 5, Do print request.

Deleting Entries Only leaf entries are allowed to be deleted, meaning that the DSA deletes a DIT entry from bottom to top. For example, to delete {CountryName US, OrganizationName 3Com, OrganizationalUnitName Engineering}, first provide {CountryName US, OrganizationName 3Com, OrganizationalUnitName Engineering} to the DSA, and the DSA will delete OrganizationalUnitName Engineering. The DIT now contains only {CountryName US, OrganizationName 3Com}. Next, have the DSA delete {CountryName US, OrganizationName 3Com}, and the DSA will delete the leaf entry OrganizationName 3Com. You then provide the entry {CountryName US} to be deleted.

To delete a name from the directory name database, follow these steps:

- 1 Select option 2, Delete name, from the Directory Manager main menu.

For example, to delete the name C=US O=3Com OU=NSD, select option 2, Delete name, from the Directory Manager main menu. The following menu is displayed:

```
-----Directory Manager Menu-----
No Default Distinguished Name
No UserName
1.- Country = US
2.- Organization = 3Com
```

- 3.- OrganizationalUnit = NSD
- 4.- ApplicationProcess
- 5.- ApplicationEntity
- 6.- Do Delete request

2 Select option 6, Do Delete request.

The name C=US O=3Com OU=NSD is deleted from the database.

Setting the User Name and Password To set up the user name and password when you do a DSA operation request, follow these steps.

1 Select option 5, Set user name and password, from the Directory Manager main menu.

The following submenu is displayed:

```
-----Directory Manager Menu-----
No Default Distinguished Name
No UserName
1.- Country
2.- Organization
3.- OrganizationalUnit
4.- Person
5.- Save user name and password
```

2 Specify Country, Organization, and OrganizationalUnit names by selecting options 1, 2, and 3, respectively.

3 Select option 4, Person.

The Common Name prompt is displayed.

CommonName:

4 Enter the user name. A maximum of 14 characters is allowed.

For example, enter the user name John. The User Password prompt is displayed:

UserPassword:

5 Enter the password.

For example, enter the password Guest.

6 Select option 5, Save user name password.

The new user name and password is displayed at the top of the screen as:

UserName: CN = John Password = Guest

Setting Up the Default Distinguished Name The following procedure describes how to set up a default distinguished name (DN) so that when you do a DSA operation request, or delete or list names, you do not have to retype certain fields of names that remain constant.

For example, the Country, Organization, and OrganizationalUnit of a directory name are often common to all devices in the same network or subnetwork. To avoid having to define them every time you access the DSA, you can specify a default name called a default DN that contains all three of them. After you define a default DN, only the parts not defined in the DN need to be defined whenever a new name is added.

The default DN must first be added to the database before it can be used as the default distinguished name for all name requests. For example, if you want to

define a DN in which US is the CountryName, 3Com is the OrganizationName, and Finance is the OrganizationalUnitName, you must first add this name to the directory. Save US3ComFinance as the default DN. You need to then specify only the Application Entity and Application Process attributes when accessing the DSA, and the default DN is automatically added to these attributes.

Follow these steps to set the name you just added to the database as the default DN. To record the default DN on the gateway only, follow these steps:

- 1 Select 6, Set Default DN, from the main menu.

The following submenu is displayed:

```
-----Directory Manager Menu-----
No Default Distinguished Name
No UserName
1.- Country
2.- Organization
3.- OrganizationalUnit
4.- Save default DN
```

- 2 Enter the name you just added to the database:

- a Select 1 from the menu and specify **us** as the country.
- b Select 2 from the menu and specify the organization by entering:
3Com
- c Select 3 from the menu and specify the organizational unit, by entering:
Finance
- d Select 4 to save the default DN and press the Return key.

The following information is displayed:

```
-----Directory Manager Menu-----
Default DN: Country=US Org=3Com OrgUnit=Finance
No UserName
1.- Country = US
2.- Organization = 3Com
3.- OrganizationalUnit = Finance
4.- Save default DN
```

The new default DN is displayed at the top of the screen.

File-Based Name Service

You can use a file-based name service in addition to, or instead of, the X.500 directory service. A file-based name service stores the name service on the gateway diskette.

Configuring the Gateway for File-Based Name Service To configure the gateway to use the file-based name service, follow these steps:

- 1 Confirm that the NameSourceOrder parameter includes file-based in its values by entering:

```
SHow -OSIAPPL NameSourceOrder
```

If you want to use both the file-based and X.500 name services, include both names and the order in which name requests should be resolved.

For example, to include both file-based and X.500 name services, and to specify that the file-based name service be queried first, enter:


```
SETDefault -OSIAPPL NameSourceOrder = File X500
```

- 2 Add names to physical addresses in the file-based name database.

For example, to assign the name "gate" to the PSAP address /47/0004/00351100080002013C3701!1.128, enter:

```
ADD -OSIAPPL Name gate /47/0004/00351100080002013C3701!1.128
```

You can use this command to add names to the database on the gateway diskette.

The name can be no more than 14 characters and must start with a letter. Characters that can follow the first letter are a letter, a digit, or one of the following symbols: underscore (_), the period (.), or the at sign (@). All other characters are ignored.

Optionally, you can delete names using the DElete -OSIAPPL Name <name> syntax.

- 3 To confirm that the new name has been successfully added, use:

```
SHow -OSIAPPL Name <name>
```

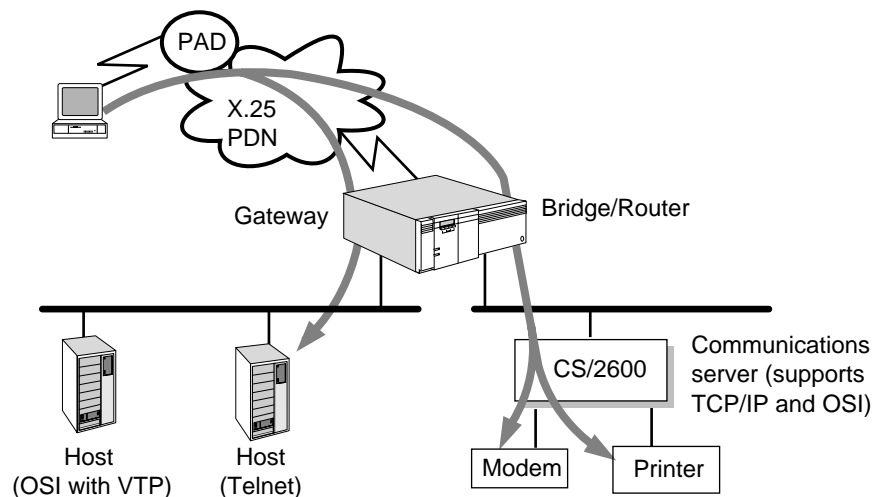
Displaying Names To display names currently stored on the gateway diskette in the filename database, enter:

```
SHow -OSIAPPL Name
```

How the Incoming Connection Service Works

The X.25 connection service gateway allows X.25 PAD-attached terminals to connect to IP Internet-attached Telnet or Rlogin hosts, OSI-based hosts, and to hosts attached to host ports on a communications server that supports Telnet, Rlogin, or the Virtual Terminal Protocol (VTP). WAN-to-LAN connections are also referred to as *incoming connections* and are controlled by the gateway's incoming connection service. Figure 407 is an example of WAN-to-LAN connections.

Figure 407 WAN-to-LAN Connections (Incoming)



The X.25 connection service gateway offers two type of incoming connections:

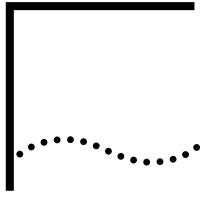
- Automatic (one-step)

End users can enter a connection command from the X.25 PAD-attached terminal, and the gateway automatically establishes the connection to the Telnet, Rlogin, or OSI server.

- Extended (two-step)

End users can enter a connection command from the X.25 PAD-attached terminal and establish a connection to the gateway user interface, the same interface that is seen when connecting to the local console of the 3Com router or connecting through Telnet or through OSI VTP. After connecting to the gateway user interface, users can make connections to Telnet, Rlogin or OSI servers. Users with Network Manager privilege can also configure, manage, and monitor the system.

With incoming automatic calls, you can connect only to a single host. With incoming extended calls, you can connect to multiple hosts and establish up to eight sessions per port.



CONFIGURING LOCAL ACCESS CONTROL

This chapter describes how to configure access control to regulate user access to the NETBuilder bridge/router. You can access the bridge/router either through a console port or Telnet connection, or through the gateway.

During incoming extended connection requests from X.25 packet assembler/disassembler (PAD)-attached terminals to Internet Protocol (IP) Internet-attached Telnet and Rlogin hosts, as well as Open Systems Interconnection (OSI)-based hosts, limiting access is crucial. Other types of access control can be implemented to prevent or restrict remote access to the gateway and to force users to log on to Rlogin servers. This chapter describes how to configure local access control, log on and log out, change user passwords, and control Rlogin connections.

Configuring Local Access Control

Local access control requires a user to specify a user name and a password before entering commands. You can use a user account name given to you by your network administrator or you can access the NETBuilder bridge/router by using the user account name, "root."

Procedure To configure access control, follow these steps:

- 1 Enable local access control, via a console port or Telnet connection, by entering:

```
SETDefault -AC RESolutionOrder = Local
```

- 2 Set the timer value used for retransmission interval.

```
SETD -ACS RetransTimer = 1
```

- 3 For X.25 PAD users to enable local access control through the gateway, enter:

```
SETDefault -AC CONTROL = Enable
```



The default setting for access control is Enable, so this parameter may already be set.

- 4 Assign a user account name by entering the AddUser command.
 - a At the prompt, enter the user's account name.
The account name is limited to 15 characters.
 - b Enter the user's full name at the next prompt.
The full user name is limited to 23 characters.
- 5 Assign the level of access by entering the user's maximum access privilege (Max. Privilege). Enter NetMgr or NM for net manager privileges or "User" or "U" for user privileges.
- 6 Enter a password for the user.

The password is limited to 15 characters and is case-sensitive. For security reasons, passwords are not echoed on the screen.

When prompted, reenter the password.

You can also use the menu-driven UserManage command to add user names and passwords. The DEleteUser command removes user accounts from the database. For more information on these commands, see the Commands chapter in *Reference for Enterprise OS Software*.

You can set other parameters that apply to local access control, such as EXpirationTimer. For more information, see the AC Service Parameters chapter in *Reference for Enterprise OS Software*.

Related Information

Local access control can prevent users from logging on to the router, but it cannot prevent users from accessing the router remotely using the REMote command. You can prevent users from accessing your router remotely, or you can restrict specific users' remote access to your router.

Logging On and Logging Out (in the CX package)

When an incoming X.25 extended connection is made to the gateway, the user must log on before entering commands if local access control is enabled.

To log on and log out, follow these steps:

- 1 Log on at the NetLogin prompt by entering the user name assigned to you by the network manager.

The gateway prompts for a password.



User account names and passwords are case-sensitive.

- 2 Enter the password assigned by the network manager.

After you enter the correct password, the following prompt is displayed:

```
Enterprise OS>
```

- 3 Log out by entering the LOGout command.

This command disconnects all sessions on a port and requires the user to log on again before entering commands.

Changing User Passwords

Users can change their passwords.



Passwords are case-sensitive, and must be entered exactly as they were assigned. If you plan to use IBM's NetView application to access the bridge/router from an MVS host, the password must be configured on the bridge/router using all upper-case letters. For information about the SNAMS service parameters that allow NetView access, see the SNAMS Service Parameters chapter in Reference for Enterprise OS Software.

To change the password, follow these steps:

- 1 Enter the PassWord command.

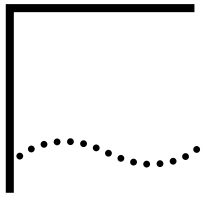
The gateway prompts you for the old password.

- 2 Enter your old password.

The gateway prompts you for your new password.

- 3 Enter your new password.

For security reasons, passwords are not displayed on the screen and cannot be viewed by the network manager.



MANAGING SESSIONS FOR INCOMING EXTENDED CALLS

This chapter describes how to make connections from the gateway's user interface to Internet Protocol (IP) Internet-attached Telnet and Rlogin servers, and to Open Systems Interconnection (OSI) hosts. When the connection has been made, you can manage the session, establish and manage multiple sessions, and disconnect sessions. It is assumed that you have already made an incoming extended connection from an X.25 packet assembler/disassembler (PAD)-attached terminal to the gateway user interface. For information on making an incoming extended connection, see the Configuring Connections for Incoming Calls chapter.

This chapter also describes procedures for making connections to different types of network resources, and provides general information on how to manage sessions, including moving between sessions and disconnecting sessions.



For conceptual information, see "Managing Sessions" later in this chapter.

Making Connections to IP Internet-attached and OSI Hosts

After establishing a connection from the X.25 PAD-attached terminal to the gateway user interface, you can use different commands to establish connections to IP Internet-attached hosts and to OSI-based hosts. The specific commands are listed in Table 95; their availability depends on the protocols being run.

Table 95 Establishing Connections to IP Internet-attached and OSI Hosts

| Setup | Step | Command |
|---------------------------|--|----------------------------|
| TCP/IP or OSI connections | Connect to a host. | Connect <name> <address> |
| TCP/IP connections | Connect to a host using the Telnet protocol. | TELnet <name> <address> |
| | Connect to a host using the Rlogin protocol. | RLOGin <name> <address> |
| OSI connections | Connect to an OSI host. | VTP <name> <psapaddress> |
| Troubleshooting | Check the status of a TCP/IP destination. | PING <address> |
| | Check the status of an OSI destination. | OPING <nsapaddress> |

The sections and procedures that follow explain the meaning of each command in detail and give examples.

Making Connections with the Connect Command

You can use the Connect command to make connections to most resource types on the network, except for Rlogin connections.

To make a connection using the Connect command, follow these steps:

- 1 Connect to the desired resource.
 - To connect to an IP address, use:
Connect <IP address>
 - To connect to an OSI PSAP address, use:
Connect <OSI PSAP address>

For more information regarding presentation service access point (PSAP) addresses and 3Com's OSI address conventions, see the NSAP and PSAP Addressing appendix.

For example, to connect to a resource named "marketing," enter:

Connect marketing

To complete the connection, the name "marketing" must be an IP name in a Transmission Control Protocol/Internet Protocol (TCP/IP) environment, or a name in an OSI environment.

The syntax for the Connect command can vary greatly, depending on the resource being accessed. For more information on the different syntax possibilities, see the Commands chapter in *Reference for Enterprise OS Software*.

You can also specify the gateway to make a connection, then immediately reenter command mode (allows you to enter other connection service commands and is indicated by the Enterprise OS prompt). To enter command mode, type the letters "ECM" after the address or name of the resource as shown in the following example:

Connect marketing ECM

If the resource you are accessing accepts the connection, the gateway sends the following message to the terminal:

```
session n -- connected to marketing
```

where *n* refers to the number of the session.

If the connection does not succeed, you will receive an error message. A connection attempt can fail for a number of reasons. For more information, see "Troubleshooting Connection Error Messages" later in this chapter. For information on the command mode and other modes of operation, see "Establishing a Single Session" later in this chapter.

- 2 When you have reached the resource, perform the actions appropriate for the resource application.

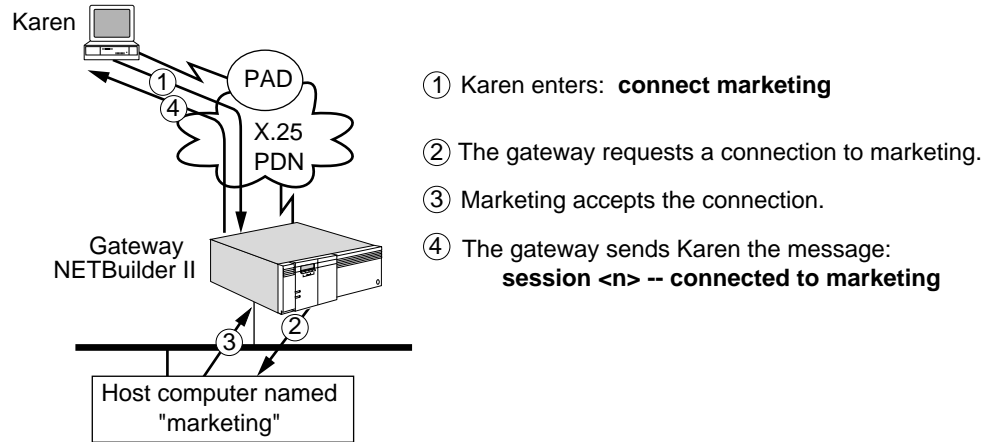
- 3 To disconnect from the session, log out from the host.

You can also enter the enter command mode (ECM) character, and then the DisConnect command, from the Enterprise OS prompt.

- 4 To disconnect from the gateway and return to the PAD terminal prompt, use the LOGout or Llisten command.

Figure 408 shows how a connection is established between a terminal user named "Karen" at the PAD-attached terminal and the host computer named "marketing."

Figure 408 Establishing a Connection with the Connect Command



Making Telnet Connections to TCP/IP Resources

The TELnet command can be used to make Telnet connections to TCP/IP resources on the network.

To make a connection with the TELnet command, follow these steps:

- 1 Connect to the desired resource.
 - To connect to an Internet address, use:


```
TELnet <Internet address>
```
 - To connect to a specific resource, use:


```
TELnet <resource name>
```

The syntax for the TELnet command can vary depending on the resource being accessed. For more information on the different syntax possibilities, see the Commands chapter in *Reference for Enterprise OS Software*.

You can also specify the gateway to make a connection, and then immediately reenter command mode (allows you to enter other connection service commands and is indicated by the Enterprise OS prompt).

To enter command mode, type the letters "ECM" after the address or name of the resource as shown in the following example:

```
TELnet finance ECM
```

The TELnet command makes a TCP connection to the specified host (or another server) using the Telnet protocol. If the resource you are accessing accepts the connection, the gateway sends the following message to the terminal:

```
session n -- connected to finance
```

where *n* refers to the number of the session.

If the connection does not succeed, you will receive an error message. A connection attempt can fail for a number of reasons. For more information, see "Troubleshooting Connection Error Messages" later in this chapter.

- 2 When you have reached the resource, perform the actions appropriate for the resource application.

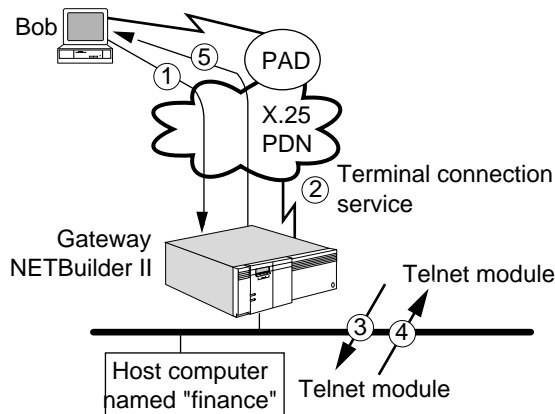
- 3 To disconnect from the session, log out from the host.

You can also enter the ECM character, and then the DisConnect command, from the Enterprise OS prompt.

- 4 To disconnect from the gateway and return to the PAD terminal prompt, use the LOGout or Llsten command.

Figure 409 shows how a connection is established using the TELnet command between a user named "Bob" at the PAD-attached terminal and a host computer named "finance."

Figure 409 Establishing a Connection with the TELnet Command



- ① Bob enters the command: **telnet finance**.
- ② The terminal connection service on the gateway looks up the name "finance" and matches an address to the name.
- ③ The Telnet module running on the gateway notifies the Telnet module running on "finance" of the connection request.
- ④ The Telnet module on "finance" tells the Telnet module on the gateway that the connection is open.
- ⑤ The gateway sends Bob a message: session <n> -- connected to finance.

Making Rlogin Connections to Resources

To make Rlogin connections, the remote host being accessed must be running Rlogin, a UNIX environment protocol. You can access remote UNIX hosts in addition to any target host that is running the Rlogin protocol.

The RLOGin command is similar to the TELnet command, but is used in a slightly different way. RLOGin differs from the TELnet command as follows:

- When making a connection with RLOGin, the client terminal always communicates the terminal type, baud rate, and user name to the host. In some cases, the client terminal may also communicate the number of rows and columns on the terminal.
- RLOGin allows the host to enable and disable flow control on the session. For example, if the client terminal is running an application where the [Ctrl]+S character has a specific meaning, the Rlogin protocol makes sure this character works in the application instead of performing the normal terminal function of [Ctrl]+S, which stops the data flow. These features are not available in all Telnet implementations.

To make a connection with the RLOGin command, follow these steps:

- 1 Connect to the desired address or resource.
 - To connect to an Rlogin resource with an Internet address, use:


```
RLOGin <Internet address>
```
 - To connect to an Rlogin resource with a specific name, use:


```
RLOGin <resource name>
```

You can also specify the gateway to make a connection, and then immediately reenter command mode (allows you to enter other connection service commands and is indicated by the Enterprise OS prompt).

To enter command mode, type the letters "ECM" after the address or name of the resource as follows:

```
RLOGin <resource name> ECM
```

If the resource you are accessing accepts the connection, the gateway sends the following message to the terminal:

```
session n -- connected to marketing
```

where *n* refers to the number of the session.

If the Rlogin connection does not succeed, you will receive an error message. A connection attempt can fail for a number of reasons. For more information, see “Troubleshooting Connection Error Messages” later in this chapter.

When Rlogin connections are made, the client user name (the user name on the client side), and the server user name (user name to be used for login on the server side) are communicated to the server during the connection negotiation. The servername is usually the same, unless you use the -l option (the letter “l”). For example, to enter an Rlogin command specifying a server user name, use:

```
RLOGin <Internet address> -l <server username>
```



Rlogin connections specifying client and server user names can affect access control. For more information on configuring access control, see the Configuring Local Access Control chapter.

- 2 When you have reached the resource, perform the actions appropriate for the resource application.

Depending on how the Rlogin host is configured, you may need to enter a password to access the Rlogin host.

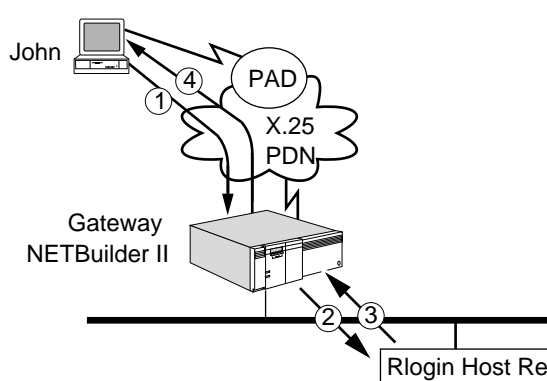
- 3 To disconnect from the session, log out from the host.

You can also enter the ECM character, and then the DisConnect command, from the Enterprise OS prompt.

- 4 To disconnect from the gateway and return to the PAD terminal prompt, use the LOGout or Listen command.

Figure 410 shows how an Rlogin connection is established between a terminal user named “John” at a PAD-attached terminal an Rlogin host named “redfiles.”

Figure 410 Establishing a Connection with the RLOGin Command



- ① John enters: **rlog redfiles**
- ② The gateway requests a connection to the Rlogin host, sending the Rlogin configuration for John's port, including:
 - Terminal type
 - Client username
 - Server username
 - Baud rate
- ③ The Rlogin host sends back confirmation of the request to the gateway. The host may request information on rows and columns from the gateway.
- ④ The connection is made to John's terminal port. If no username is sent in the connection request, the user must enter a password to access the remote host.

Making Connections to OSI Resources

You can use the VTp command to connect to OSI resources on the network. When the VTp command is specified, an OSI connection is made to a specified name or PSAP address. If a list of addresses or names is entered, the gateway tries one address or name after another in the given order until a connection is made.

To make an OSI connection, follow these steps:

- 1 At the Enterprise OS prompt, enter VTp, followed by the name or address to which you want to connect.

For example, to connect to an OSI resource named "bluefiles" shown in Figure 411, enter:

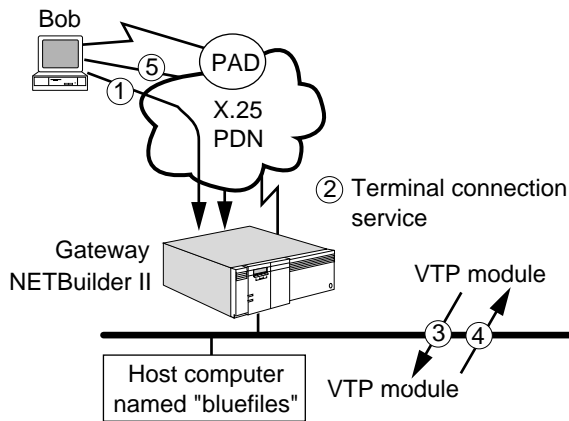
VTp bluefiles

If you do not know the name of a destination, or a name has not been defined, you can specify the destination NSAP address followed by the upper layer addresses. On a 3Com terminal server, a selector is used to specify a port number. Figure 412 shows how a PAD-attached user connects to the modem attached to port 6 on Server B (NSAP address /47/0005/01ABCDEF000000100030080002056821900) by entering:

VTp /47/0005/01ABCDEF000000100030080002056821900!6

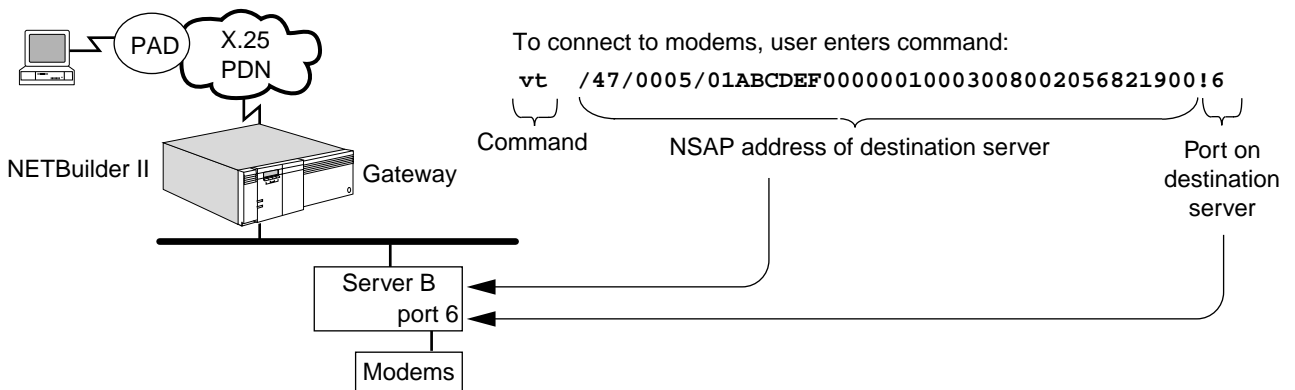
In this example, !6 is the selector. For information on selectors, see the NSAP and PSAP Addressing appendix.

Figure 411 Connecting to a Name



- 1 Bob enters the command: vt bluefiles
- 2 The terminal connection service on the gateway looks up the name "bluefiles" and matches an address to the name.
- 3 The VTP module running on the gateway notifies the VTP module running on "bluefiles" of the connection request.
- 4 VTP module on "bluefiles" tells the VTP module on gateway that the connection is open.
- 5 The gateway sends Bob this message: session <n> -- connected to bluefiles

Figure 412 Connecting to an OSI Address



You can also specify the gateway to make an OSI connection, then immediately reenter command mode (allows you to enter other connection service commands and is indicated by the Enterprise OS prompt). Type the letters "ECM" following the address or name of the resource as shown in the following example:

VtP bluefiles ECM

If the connection does not succeed, you will receive an error message. There are different reasons why a connection attempt can fail. For more information, see "Troubleshooting Connection Error Messages." For information on the command mode and other modes of operation, see "Establishing a Single Session" later in this chapter.

- 2 When you have reached the resource, perform the actions appropriate for the resource application.
- 3 To disconnect from the session, log out from the host.
- 4 To disconnect from the gateway and return to the PAD terminal prompt, use the LOGout or Listen command.

You can also enter the ECM character, and then the DisConnect command from the Enterprise OS prompt.

Troubleshooting Connection Error Messages

Connection attempts can fail for different reasons, ranging from errors in the command syntax to discrepancies in how resources are configured. Also, a specified resource may not be configured, or it may not be reachable because of a problem on the network. This section lists the meaning of some common error messages.

Connecting using IP ... aborted, no response from remote host

Meaning: The destination host did not respond to the connection request. The host could be down, or no path exists on the network to this host. The gateway sends this message when it does not receive a response from the host within a given time.

Action: None.

Connecting using IP ... Terminated by the remote TCP host, Reset received

Meaning: The destination host responded to the connection request with a reset packet, in effect, refusing the connection.

Action: None.

IPName: No adequate response received

Meaning: The gateway did not resolve the Internet name entered in the connection request. The name does not exist on the network, or the name was not entered correctly.

Action: Verify that the correct name (including upper- and lowercase letters) was entered.

No more sessions for this port

Meaning: The maximum amount of active sessions for the port has been reached.

Action: To change the number of sessions allowed on each port, the network manager uses the SETDefault !<configfile> -TERM MaxSessions syntax. A single port cannot run more than eight incoming sessions at a time.

X.500 has been selected in -OSI NameSourceOrder but DSAAddress is not configured

Meaning: The gateway did not find the OSI name or address specified. This message can also appear if a non-OSI connection attempt fails; depending on how the -DIR RESolutionOrder parameter is configured, the gateway tries an OSI connection if the non-OSI connection attempt fails. (For example, if an Internet address is entered, an OSI host will not recognize it.)

Action: None.

Checking Network Resources

If you have difficulty connecting to a network resource, you can check to see if the network resource is alive. This procedure differs if you are trying to connect to a TCP/IP resource or an OSI resource; see the appropriate sections below.

Checking TCP/IP Network Resources

If a Telnet or Rlogin connection attempt fails, or if you are not sure if a network resource is available for TCP/IP connections, you can check to see if the resource is "alive," or able to accept a connection by using the PING command.

You can ping a resource with an Internet address using:

```
PING [!<source port> | !<source-IP>][C<Count>][W<Wait>]
  [T<TTL>][L<Length>][I<Increment>][D<"Data">][Record] <target-IP |
  target-name>
```

After you enter the command, the gateway sends a request to the target resource to see if it is alive (and available for a connection). If so, the gateway provides an acknowledgment. For example, with an Internet address of 129.213.202.115 the following message is displayed:

```
ping 129.213.202.115
Pinging 129.213.202.115, source 129.213.202.111 (! 1)

129.213.202.115 is alive: time = 1 ms, seq = 439
***Success rate is 100 percent, round-trip min/avg/max = 1/1/1 ms
```

If the target resource is alive, you can then make a connection. If the target resource is not alive, or the gateway cannot find the target resource, you will receive one of several possible error messages depending on the problem. For example, if the resource is on the network, but is not alive, you will receive a message similar to the following message:

```
ping 129.213.202.115
Pinging 129.213.202.115, source 129.213.202.111 (! 1)

No reply from 129.213.202.115 , Request timed out: seq = 440
***Success rate is 0 percent
```

Other problems can cause a lack of response. For example, the target resource may be on the network, but cannot respond because of a configuration or a hardware problem. Or, the name of the resource entered may be entered incorrectly, or may not exist on the network.

You also may not get a response if there is no route configured to the IP address. If no route exists, a message similar to the following is displayed:

```
ping 129.213.202.115
129.213.202.115 is unreachable, No local route
```

This message indicates that the gateway cannot reach the address, either because the address does not exist on the network or a route has not been configured to reach that address. You can receive this message if you enter the address incorrectly. For example, if you enter the address 129.14.8.36 when the correct address is 129.41.8.36, the gateway will not find the subnet because the subnet numbers are transposed.

Checking OSI Network Resources

If a VTp connection attempt fails, or if you are not sure if a network resource is available for OSI connections, you can check to see if the resource is "alive," or able to accept a connection. Use the OPING command to see if an OSI network resource is alive.

You can ping a resource with an network service access point (NSAP) address by using the following OPING syntax which includes the NSAP address:

```
OPING <NSAP address>
```

After you enter the command, the gateway sends a request to the target resource to see if it is alive (and available for a connection). If so, the gateway provides an acknowledgment.

For example, with an NSAP address of /47/0004/0035110008000201F00801 the following message is displayed:

```
oping /47/0004/0035110008000201F00801
pinging . . . destination is alive
```

If the target resource is alive, you can then make a connection. If the target resource is not alive, or the gateway cannot find the target resource, you will receive one of several possible error messages depending on the problem. For example, if the resource is on the network, but is not alive, a message similar to the following is displayed:

```
oping /47/0004/0035110008000201F00801
pinging . . . dest unreachable according to local routing table
```

Other problems can cause a lack of response. For example, the target resource may be on the network, but cannot respond because of a configuration or a hardware problem. Or, the name of the resource entered may be entered incorrectly, or may not exist on the network.

You also may not get a response if there is no route configured to the NSAP address. If no route exists, a message similar to the following is displayed:

```
pinging. . .received Error Report PDU code 128
```

This message indicates that the gateway cannot reach the address, either because the address does not exist on the network or a route has not been configured to reach that address. You can receive this message if you enter the address incorrectly.

If you still get no response, you can use the OTraceRoute command to trace a path to an OSI destination. For example, to trace the path to the NSAP address above, you would enter:

```
OTraceRoute /47/0004/0035110008000201F00801
```


This command will then display the path to the destination, if it can be found. For more information on the OTraceRoute command, see the Commands chapter in *Reference for Enterprise OS Software*.

Managing Sessions

A *session* is a logical connection between two devices through one or more gateways. A session usually is initiated from a terminal at one end of the connection. Sessions also can be initiated by a network manager on either a local or a remote station as described in the Network Management chapter.

Table 96 summarizes the session management commands that can be used after incoming extended connection session establishment.

Table 96 Session Management Commands

| Setup | Step | Command |
|--------------------------------|--|--|
| Establish a single session | Enter a connection command. | Connect <name> <address>
TELnet <name> <address>
RLOGin <name> <address>
VTp <psapaddress> <name> |
| Establish multiple sessions | Enter "ECM" after connection command. | Connect <name> <address> ECM
TELnet <name> <address> ECM
RLOGin <name> <address> ECM
VTp <psapaddress> <name> ECM |
| Move between sessions | Resume the current session. | RESume |
| | Resume a session other than the current one. | RESume <sessionnumber> |
| | Resume the session number following the current session. | FORwards |
| | Resume the session number preceding the current session. | BACKwards |
| | Change the current session. | SWitch <sessionnumber> |
| Display sessions | Show all sessions on an active port. | SHow -TERM SESsions |
| | Show all sessions on the gateway. | SHow -TERM AllSessions |
| Change modes of operation | Change from command to listen mode. | LIsTen |
| | Change from data transfer mode to command mode. | Enter ECM character (default is [Ctrl]+[Shift]+6) |
| Disconnect session connections | Disconnect the current session. | DisConnect |
| | Disconnect a session other than the current one. | DisConnect <sessionnumber> |

The sections that follow describe each of these commands in detail and give examples.

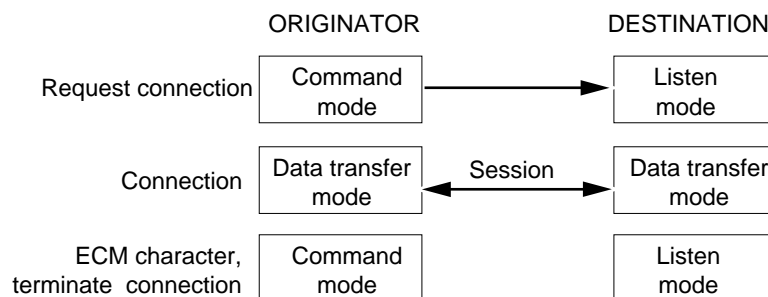
Establishing a Single Session

To establish a single session, enter a connection command (for example Connect, TELnet, RLOGin, or VTp) in command mode. Command mode is indicated by the command prompt. The default command prompt is Enterprise OS > at User privilege level and Enterprise OS # at Network Manager privilege level. During connection establishment, the gateway selects the next available port; the port number can be from 0 to 127 on a NETBuilder II system. After the gateway

establishes the connection, the selected port at which you enter commands is in data transfer mode, and is actively communicating with a destination.

Figure 413 shows the difference among command mode, data transfer mode, and the inactive state of listening mode.

Figure 413 Different Port Modes



Establishing Multiple Sessions

You can hold more than one session at a time during incoming extended connections, but you are limited to one session incoming automatic connections. Use the `SHow -TERM MaxSessions` command to determine the maximum number of sessions that you can hold simultaneously.

If you are already connected to a resource and want to initiate another session, you must enter the ECM command option to switch from data transfer to command mode. Alternatively, you can specify the ECM option when entering the Connect command to restore the port to command mode. You then initiate another session by entering the appropriate connection command.

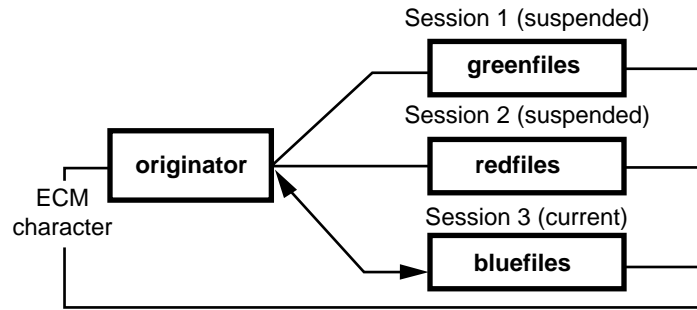
For example, the following commands are entered to establish three sessions from the connection service gateway:

```
Connect greenfiles ECM
Connect redfiles ECM
Connect bluefiles
```

The first command establishes a session with a resource named "greenfiles" and leaves the port in command mode. The second command establishes a session with a resource named "redfiles" and leaves the port in command mode. The third command establishes a session with "bluefiles" and places the port in data transfer mode. The session with "bluefiles" is called the current session. Only one session at a time is active; all other sessions are suspended and flow-controlled. The sessions are numbered sequentially as you create them. Figure 414 illustrates the concept of multiple sessions on one port.

When you specify ECM in a gateway command type the letters "ECM"; when specifying the ECM character from an actual session press the key combination `[Ctrl]+[Shift]+6`. For more information on the ECM character, see "Using the ECM Character to Enter Command Mode" later in this chapter.

Figure 414 Multiple Sessions



Displaying Session Information

To display a numbered list of sessions on your PAD-attached terminal, follow these steps:

- 1 If the terminal is currently in a session, enter the ECM character to restore the port to command mode.

The default is [Ctrl]+[Shift]+6.

- 2 To display active sessions on the port you are currently using for connections, enter:

SHoW -TERM SESSions

The following display appears:

```

Sessions on Portid !1
Port/session# state/protocol      Td cnt   Rd cnt
! 1/3 CNCTD/OSI TO bluefiles      0         0
! 1/2 CNCTD/TCP TO redfiles       0         0
! 1/1 CNCTD/TCP TO greenfiles     0         0
  
```

The first session listed in the display is always the current session. For example, in the preceding display, the session with "bluefiles" is the current session.

Changing the Current Session

To change the current session, follow these steps:

- 1 Enter the ECM character.
The default is [Ctrl]+[Shift]+6.
- 2 Switch to session 2 by entering: **switch 2**

Session 2 with "redfiles" is now the current session. To confirm that the current session is now session 2, enter:

SHoW -TERM SESSions

This command displays active sessions on the port you are currently using for connections.

Moving between Sessions

Use the RESumE, FORwards, and BACKwards commands to move between sessions.

Using the RESume Command

To resume the current session from the gateway command mode, enter the RESume command. For example, to resume the session with "redfiles" in the previous example, follow these steps:

- 1 At the Enterprise OS prompt, enter:

```
RESume
```

- 2 To resume a session other than the current one, enter the RESume command followed by the session number.

If you are unsure of the number of the session you want to resume, enter:

```
SHow -TERM SESSions
```

Using the FORwards and BACKwards Commands

Use the FORwards command to resume the session number following the number of your current session and the BACKwards command to resume the session number preceding the number of the current session. For example, if you have multiple sessions on port 1 and you enter the SHow -TERM SESSions command, the following display appears:

```
Sessions on Portid !1
Port/session# state/protocol      Td cnt      Rd cnt
! 1/2 CNCTD/TCP TO redfiles       0           0
! 1/3 CNCTD/TCP TO greenfiles     0           0
! 1/1 CNCTD/OSI TO bluefiles      0           0
```

In this example, session 2 is the current session. If you enter the FORwards command, session 3 is resumed. If you enter the BACKwards command, session 1 is resumed.

Using the ECM Character to Enter Command Mode

After you have established a session, the gateway is in data transfer mode, and the gateway commands are not accessible. To establish other sessions, disconnect sessions, or enter other gateway commands, you must exit the current session using the ECM character.

The default ECM character is the key combination [Ctrl]+[Shift]+6, which produces a double caret (^ ^). On most standard keyboards, the caret (^) is the same as the key for the number 6; if the caret is on a different key, press the Control key and the appropriate caret key.

You can change the default ECM character when the application requires that the double caret character (^ ^) be transmitted as data. You can change the default ECM character for all your new sessions on the gateway using the SETDefault -TERM ECMChar command. If you change the default ECM character, it only affects new sessions; for sessions already existing, the previous ECM character is still required. For example, to change the ECM character for the gateway to the key combination [Ctrl]+T (^T), enter:

```
SETDefault -TERM ECMChar = '^T'
```

You can also change the ECM character for only the current session with the SET -TERM ECMChar command. The change affects only the current session and does

not affect sessions on other ports. For example, to change the ECM character for a session on an active port to the key combination [Ctrl]+T (^T), enter:

```
SET -TERM ECMChar = `^T`
```



When you exit a session with the ECM character, you are not disconnecting the session. For more information about disconnecting sessions, see the next section.

Some applications require that the ECM character be disabled because the ECM escape characters are interpreted as normal data. In such cases, the BReakAction parameter can be configured so that entering the Break key causes the gateway to enter command mode (for example, setting BReakAction to "EscDTM"). For more information, see the TERM Service Parameters chapter in *Reference for Enterprise OS Software*

The ECM escape character is used only to enter command mode from *within* an active session. By typing the letters "ECM" at the end of a connection command, you can instruct the gateway to make a connection, then automatically reenter command mode. For example, to connect to a resource named "greenfiles," and then enter command mode, enter:

```
Connect greenfiles ECM
```

Disconnecting a Single Session

To disconnect a session, enter:

```
DisConnect
```

The port is in command mode.

Disconnecting Multiple Sessions

When holding more than one session, to disconnect the current session enter:

```
DisConnect
```

To disconnect a session other than the current one, use the DisConnect command followed by the session number. For example, to disconnect the session with "redfiles" (shown in Figure 414), enter:

```
DisConnect 2
```

The DisConnect command leaves your port in command mode. To disconnect all PAD sessions and the X.25 connection, and place the port in listening mode, enter:

```
Listen
```

Changing Session Parameters

To change session parameters for the current session, use:

```
SET -TERM <parameter>
```

To change your session parameters for future sessions, use:

```
SETDefault -TERM <parameter>
```

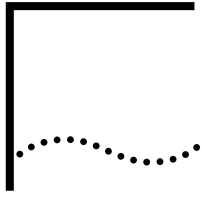
To view session-related parameters that can be changed, enter:

```
SHow -TERM DefaultParams
```



You must have an active port to display the default parameter values that initialize the port and session.

For a list of parameters affecting sessions, see the TERM Service Parameters chapter in *Reference for Enterprise OS Software*.



CONFIGURING INTERNETWORKING USING ATM

This chapter describes how to configure your NETBuilder II bridge/router to establish LAN, WAN, and MAN connectivity through Asynchronous Transfer Mode (ATM).



For conceptual information, see “How ATM Works” later in this chapter.

The bridge/router supports both bridging and routing of multiple protocols over ATM. The ATM Service allows your bridge/router to transmit and receive data over a permanent virtual circuit (PVC) with any other device on the ATM network. You can achieve multiprotocol encapsulation over ATM through PVCs by upgrading to software version 9.0 or higher and installing the MP ATMLink module in your NETBuilder II bridge/router. In this configuration, your bridge/router supports operation over ATM adaptation layer 5 (AAL5) and router cluster topologies in meshed, partially meshed, and nonmeshed topologies.

Setting Up the ATM Service

This section describes how to configure your bridge/router to transmit and receive data over an ATM interface using PVCs with the following protocols:

- Transparent bridging
- Source Route bridging
- IP routing
- IPX routing

You must follow the steps in this section whether you are configuring for bridging or for routing. After you have completed these steps, see the appropriate section:

- “Configuring Transparent Bridging”
- “Configuring Source Route Bridging”
- “Configuring IP Routing”
- “Configuring IPX Routing”

For detailed descriptions of all commands, see *Reference for Enterprise OS Software*.

Prerequisites

Before beginning this procedure, complete the following tasks:

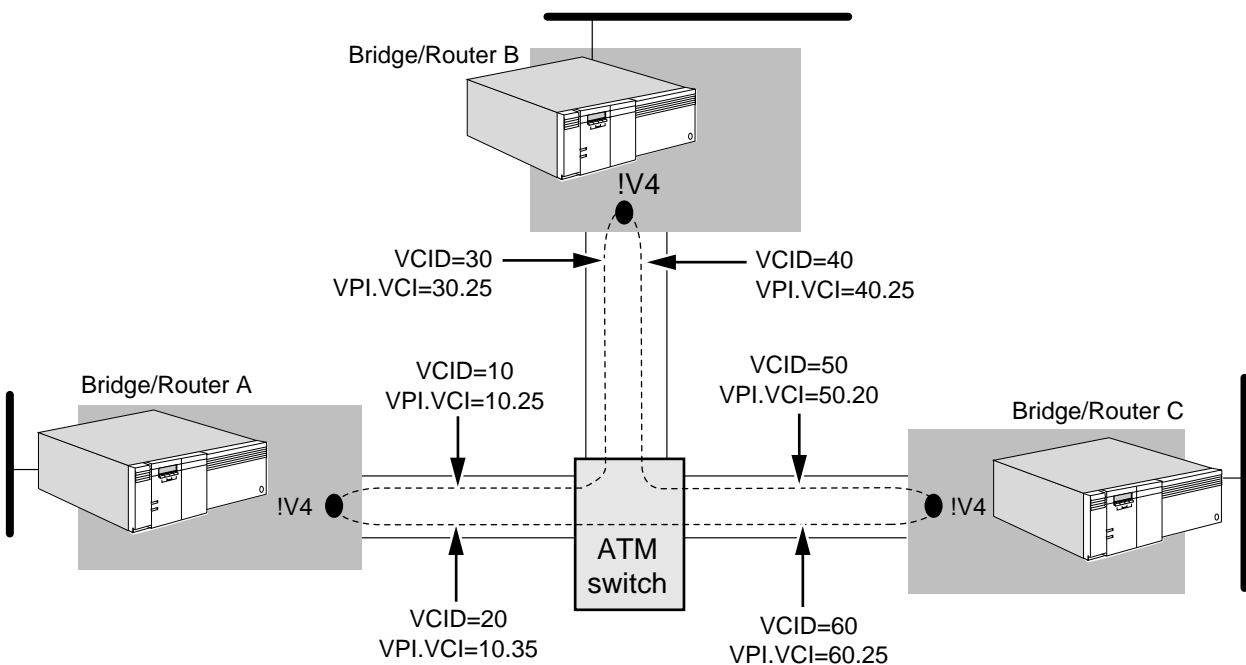
- Log on to the system with Network Manager privilege.
- Configure your bridge/router ports, virtual ports, and paths according to the Configuring Basic Ports and Paths chapter and the Configuring Advanced Ports and Paths chapter.

- Obtain the ATM addresses (VPI.VCI) from your ATM service provider or the ATM switch.
- Determine if you have a partially meshed or nonmeshed topology.

If you plan to enable the Internet Protocol-Routing Information Protocol (IP-RIP) or the Internetwork Packet Exchange (IPX) routing protocol, you need to make certain that the next-hop split horizon feature is enabled. If you have a partially meshed or nonmeshed topology, and you plan to enable Open Shortest Path First (OSPF), make sure that you set `-OSPF CONTrol` to `NonMesh` to enable the point-to-multipoint interface. For information on meshed, partially meshed, and nonmeshed topologies, next-hop split horizon, and virtual ports, see "How ATM Works" later in this chapter. For instructions on setting up virtual ports, see the *Configuring Advanced Ports and Paths* chapter.

Procedure To transmit and receive data over an ATM network, see Figure 415 and follow these steps on both ends of the link:

Figure 415 Configuring the ATM Service



- 1 Verify that the port owner setting is ATM by entering:

```
SHow -PORT OWNer
```

The NETBuilder II bridge/router automatically sets the port owner to ATM if the MP ATMLink module is installed. If the setting for the port is not correct, use the `SETDefault` command.

For example, to set the owner on port 4 to ATM on bridge/router A, enter:

```
SETDefault !4 -PORT OWNer = ATM
```

- 2 Create a virtual port for each remote network that is attached to the ATM network using:

```
ADD !<port> -PORT VirtualPort <path> MPATM
```


For example, to configure a virtual port for path 4 on bridge/router A, enter:

```
ADD !V4 -PORT VirtualPort 4 MPATM
```

Enter similar commands on bridge/routers B and C.

Each ATM virtual port has a unique media access control (MAC) address, and virtual ports are limited to 64 per physical interface.

- 3 For outbound traffic, configure a traffic shaper to control the traffic flow using:

```
SETDefault -ATM TrafficShaper = <id>(1-14) <peak>(1-155,000) <avg>(1-155,000) [<burst>(1-255)] [High | Low]
```

Based on the user applications, configure the traffic shaper options. Only AAL5 data-application traffic (not voice and video application traffic) is supported.

- a Supply the ID of the shaper to be modified. Valid IDs are from 1 to 14.
- b Specify the peak rate and average rate in kilobits per second. Valid rates are from 1 to 155,000.
- c Specify the burst count in 53-byte cells. Valid numbers are from 1 to 255. The default burst count is 32.
- d Specify the priority level. Valid priorities are High or Low.
- e Virtual circuit traffic associated with a high-priority shaper are serviced first. The default priority is High. If several traffic shapers have the same priority, they are serviced in a round-robin process and considered to be equal priority.

For example, to configure shaper 9 with a 10 kbps peak rate, an 8 kbps average rate, a burst count of 64 53-byte cells, and a high priority, enter:

```
SETDefault -ATM TrafficShaper = 9 10 8 64 High
```

For conceptual information about traffic shaping, see “Quality of Service” and “Traffic Shapers” later in this chapter.

- 4 Add a permanent virtual circuit for the virtual port, and map its unique virtual circuit identifier (VCID) to the service provider’s VPI.VCI using:

```
ADD !<port> -ATM PermVirCircuit <vcid> <vpi.vci> [LLCSNAP | [NULL | IP | IPX]] [<shaper_id>]
```

- a Supply a VCID between 1 and 1024; enter the VPI.VCI number supplied by the ATM service provider.
- b Supply an encapsulation type. Use LLCSNAP to allow multiple protocol types to be carried within a single ATM virtual circuit. Use NULL and the keyword IP or IPX when only one protocol is configured to run on the virtual circuit.
- c Select a traffic shaper ID between 1 and 14 for outgoing traffic that was previously configured in step 3.

For example, to assign VCIDs of 10 and 20 to the VPI.VCIs of 10.25 and 10.35 on virtual port !V4 with LLCSNAP encapsulation using traffic shaper 9 on bridge/router A, enter:

```
ADD !V4 -ATM PermVirCircuit 10 10.25 LLCSNAP 9
```

```
ADD !V4 -ATM PermVirCircuit 20 10.35 LLCSNAP 9
```

Enter similar commands on bridge/routers B and C, making sure to use the same encapsulation type.

- 5 If necessary, adjust the size of the VPI and VCI bits to match the size supported by the ATM switch using:

```
SETDefault !<port> -ATM VPIBits = <vpi_bits>(1-8)
SETDefault !<port> -ATM VCIBits = <vci_bits>(1-16)
```

By default, VPI is set to 6, and VCI is set to 10. Valid VPI numbers range from 0 to 255, valid VCI numbers range from 0 to 65,535 when the full range of bits is used. VPI.VCIs from 0.0 to 0.32 are reserved virtual circuits and are not allowed as user virtual circuits. The VPI and VCI values must be compatible with the configured value for the VPIbits and VCIBits parameters.

- 6 If you adjust the VPIBits and VCIBits parameters, re-enable the path using:

```
SETDefault !<path> -PATH CONTROL = Enabled
```

Verifying the Configuration

To verify your ATM configuration, follow these steps:

- 1 Display current ATM configuration information by entering:

```
SHow -ATM CONFIguration
```

Verify that your ports and paths, and the PVC are correctly configured.

- 2 Obtain ATM distributed protocol module (DPM) statistics using:

```
SHow -SYS [!<port | slot>] DpmSTATistics [POrt | SLOt] [SRc | DESt]
[<SUMmary | ALl | BRIdge | IP>]
```

This display shows per-slot or per-port statistics for IP or bridge data sent or received on the ATM interface.

- 3 Obtain virtual ports statistics for IPX by entering:

```
SHow -SYS STATistics -IPX
```

This display shows IPX per port statistics for data sent or received over ATM virtual ports.

For detailed statistic information, see the Statistics Displays appendix.

Monitoring the Network

If you are experiencing connectivity problems, monitor the virtual circuit and the network connectivity status by following these steps:

- 1 Specify the time interval at which the interface is checked to determine whether it is connected to the ATM network using:

```
SETDefault !<port> -ATM KeepAliveTime = <seconds>(1-60)
```

The default setting is 2 seconds.

- 2 Re-enable the path to make the changes to the KeepAliveTime parameter effective using:

```
SETDefault !<path> -PATH CONTROL = Enabled
```

- 3 Determine if the interface is connected to the ATM network using:

```
SETDefault !<port> -ATM LoopMode = AssumeConnected | DetectFraming |
LoopBack
```

The DetectFraming option determines whether the interface is connected to the ATM network if successful framing of received data has occurred.

The LoopBack option determines whether the interface is connected to the ATM network if F4 loopbacks to the ATM switch are successful.

- 4 Obtain end-to-end connection status by performing end-to-end loopback testing for all virtual circuits associated with the specified virtual port using:

```
SETDefault !<port> -ATM VirCirLoopTime = <seconds>(1-60)
SETDefault !<vport> -ATM VirCirLoopMode = ENabled
```

By default, the VirCirLoopTime parameter is set to 5 seconds. It specifies the time interval in seconds to initiate the F5 loopback to determine the end-to-end connection status.

5 Check ATM connection manager diagnostics using:

```
SHow -DIAGnostic ATM [ALL | ConnectionTable <slot number:1-8> | ADDR]
```

For more information about the display for this parameter, see the DIAGnostic Service Parameters chapter in *Reference for Enterprise OS Software*.

Configuring Transparent Bridging

This section describes how to configure transparent bridging over ATM using PVCs.

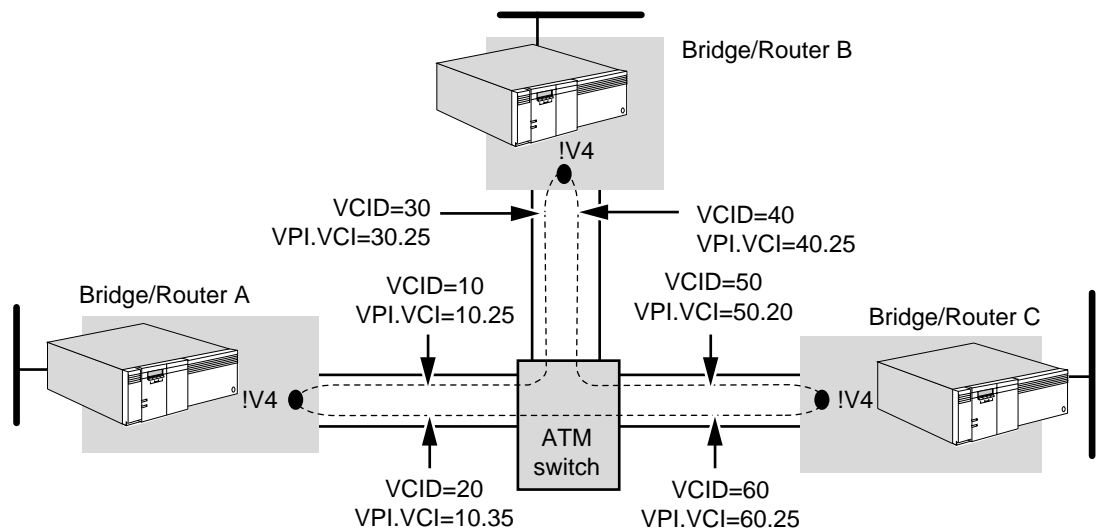
Prerequisites Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the transparent bridging instructions in the Configuring Bridging chapter.
- Set up the ATM Service as described in “Setting Up the ATM Service” earlier in this chapter.
- Obtain the virtual circuit identifier (VCID) of the PVCs for each bridge/router participating in transparent bridging.

Transparent bridging does not correctly operate in some nonmeshed topologies. For more information, see “Fully Meshed, Partially Meshed, and Nonmeshed Topologies” later in this chapter.

Procedure To configure transparent bridging over ATM, see Figure 416 and follow these steps:

Figure 416 Configuring Transparent Bridging over ATM



- 1 Verify that transparent bridging is enabled on each bridge port that is directly connected to the ATM switch.

By default, transparent bridging is enabled. To verify the setting, on each device use:

```
SHow -BRidge TransparentBRidge
```

If transparent bridging has been disabled, you can enable it on virtual port 4 of bridge/routers A, B, and C. On each of these devices enter:

```
SETDefault !V4 -BRidge TransparentBRidge = TransparentBRidge
```

- 2 Enable the bridge by entering:

```
SETDefault -BRidge CONTROL = Bridge
```

- 3 Specify the local VCID of the PVCs connecting to bridge neighbors that are participating in bridging over ATM.

For example, to specify bridge/router A's local VCIDs of the PVCs connecting to bridge/routers B and C and map them to virtual port !V4, on bridge/router A enter:

```
ADD !V4 -BRidge ATMNeighbor = 10
```

```
ADD !V4 -BRidge ATMNeighbor = 20
```

Enter similar commands on bridge/routers B and C to configure ATM for their neighbors. You can configure up to 256 neighbors on a virtual port.

This completes the procedure for configuring bridging over an ATM switch.

Configuring Source Route Bridging

This section provides information for configuring source route bridging over ATM.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in the Configuring Source Route Bridging chapter.
- Set up the ATM service as described in "Setting Up the ATM Service" earlier in this chapter.
- Assign a unique ring number for each remote network.
- Assign a bridge number for the bridge.

Procedure

To configure source route bridging over ATM, follow these steps:

- 1 Assign each wide area port of each bridge/router that is attached to the ATM network the ring number (hexadecimal) of the network it accesses using:

```
SETDefault !<port> -SR RingNumber = <number> (1-4095) | 0x<number> (1-FFF)
```

You can enter the ring number in decimal or hexadecimal format. Precede the hexadecimal number with 0x.

- 2 Verify that source route bridging is enabled on the wide area port using:

```
SHow !<port> -SR SrcRouBridge
```

If source route bridging is disabled, you need to enable it for your wide area port using:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```

- 3 If you want to run source route and transparent bridging on a NETBuilder II bridge/router, skip this step and go on to step 4. If you want to run source route bridging only on a NETBuilder II bridge/router, disable transparent bridging on the wide area port using:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

This step does not apply to model 32x and 52x SuperStack II bridge/router. Transparent bridging is not supported on these models.

- 4 Specify the local VCID of the PVCs connecting to bridge neighbors that are participating in bridging over ATM using:

```
ADD !<port> -BRidge ATMNeighbor = <VCID>
```

This completes the procedure for configuring source route bridging over an ATM switch.

- 5 Verify that bridging is enabled by entering:

```
SHow -BRidge CONFIguration
```

If bridging has been disabled, enable it for the system by entering:

```
SETDefault -BRidge CONTRol = BRidge
```

Configuring IP Routing

This section describes how to configure IP routing over ATM using PVCs.

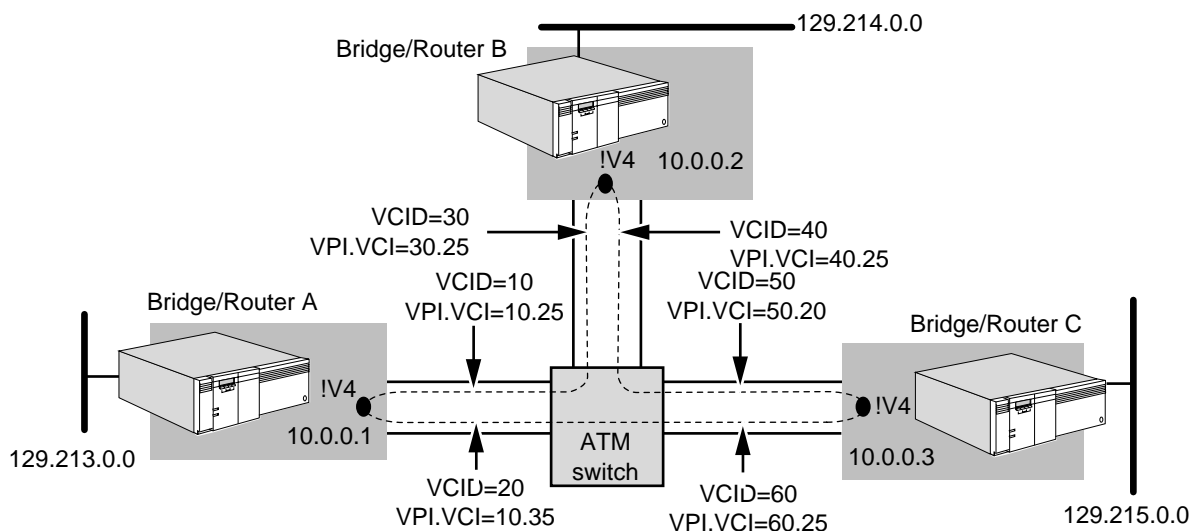
Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure your LAN according to the procedures in the Configuring IP Routing chapter.
- Set up the ATM Service as described in "Setting Up the ATM Service" earlier in this chapter.
- Determine the IP addresses for each port of your bridge/router that is attached to the ATM switch.
- Obtain the IP address and the VCID of the PVCs for each bridge/router that is attached to the ATM switch and participating in IP routing.

Procedure To enable IP to operate over an ATM switch, see Figure 417 and follow these steps:

Figure 417 Configuring IP over ATM



- 1 Assign an IP address to each virtual port on each bridge/router that is directly attached to the ATM switch.
For example, the following command assigns the address 10.0.0.1 to virtual port 4 on bridge/router A:

```
SETDefault !V4 -IP NETaddr = 10.0.0.1
```

- 2 Specify IP-to-ATM address mapping information for all neighbors.

The following sequence of commands specifies IP-to-ATM address mapping information for the bridge/routers directly attached to the ATM switch. In the examples that follow, ATM can be used in place of the and sign (&) when specifying the VCID of the PVC. The VCID has local significance and is mapped to the PVC associated with the neighbor.

For example, on bridge/router A (IP address 10.0.0.1) enter:

```
ADD -IP ADDRESS 10.0.0.2 &10
```

```
ADD -IP ADDRESS 10.0.0.3 &20
```

Enter similar commands on bridge/router B (IP address 10.0.0.2) and bridge/router C (IP address 10.0.0.3), specifying the IP address and the local VCID.

- 3 Enable the dynamic routing protocols using Routing Information Protocol-Internet Protocol (RIP) or Open Shortest Path First (OSPF) for each virtual port.

- To learn routes dynamically on virtual port 4 using RIP, determine if the ATM network is fully meshed or partially meshed. If it is fully meshed, then enter:

```
SETDefault !V4 -RIP CONTROL = (TALK, Listen, FullMesh)
```

If it is partially meshed, enter:

```
SETDefault !V4 -RIP CONTROL = (TALK, Listen, NonMesh)
```

Setting the CONTROL parameter to the TALK and Listen values allows the router to send and receive routing information with other routers using RIP.



The *RIP Service CONTROL* parameter enables or disables RIP routing for the specified port. Neighbor learning is enabled by default (*DynamicNbr*) which causes new addresses to be learned through the Inverse Address Resolution Protocol (*InARP*) and dynamically updates the *RIP AdvToNeighbor* list. If *NoDynamicNbr* is specified, *RIP's AdvToNeighbor* list is not updated with new addresses and the neighbors list must be manually configured.

- To enable routes dynamically on virtual port 4 using OSPF, determine whether the ATM network is fully meshed or partially meshed.

If the network is fully meshed, enter:

```
SETDefault !V4 -OSPF CONTROL = (Enable, FullMesh)
```

If the network is partially meshed, enter;

```
SETDefault !V4 -OSPF CONTROL = (Enable, NonMesh)
```

All of the OSPF neighboring routers must be configured with the same mode: FullMesh or NonMesh. NonMesh is the default setting for this parameter.

After OSPF operation has been enabled, the router exchanges routing information with other routers using OSPF.



The *OSPF Service CONTROL* parameter enables or disables OSPF routing for the specified port. Neighbor learning is enabled by default on nonbroadcast multi-access (NBMA) interfaces, which means that neighbor lists are automatically created and OSPF operates correctly without static neighbor information. Neighbor learning can be disabled (*NoDynamicNbr*) for security reasons so that only those statically configured neighbors exchange routing information.

- 4 If the port is configured with neighbor learning disabled, manually specify neighbors for the routing protocols.
 - a If your network is running RIP, specify a list of neighbor addresses to which RIP will send update packets.

For example, on bridge/router A, add the IP addresses of neighboring bridge/routers B and C, enter:

```
ADD !V4 -RIP AdvToNeighbor 10.0.0.2
ADD !V4 -RIP AdvToNeighbor 10.0.0.3
```

Enter similar commands on bridge/routers B and C.
 - b If your network is running OSPF, specify a list of neighbor addresses to which OSPF will send update packets.

For example, on bridge/router A, add the IP addresses of neighboring bridge/routers B and C, enter:

```
ADD !V4 -OSPF Neighbor 10.0.0.2
ADD !V4 -OSPF Neighbor 10.0.0.3
```

Enter similar commands on bridge/routers B and C.
- 5 Verify that IP routing is enabled on each bridge/router that is attached to the ATM switch by entering:


```
SHOW -IP CONFIGURATION
```

If IP routing has been disabled, enable it by entering:

```
SETDefault -IP CONTROL = ROute
```

This completes the procedure for configuring IP routing over an ATM switch.

Configuring IPX Routing

This section describes how to configure IPX routing over ATM using PVCs.

Prerequisites

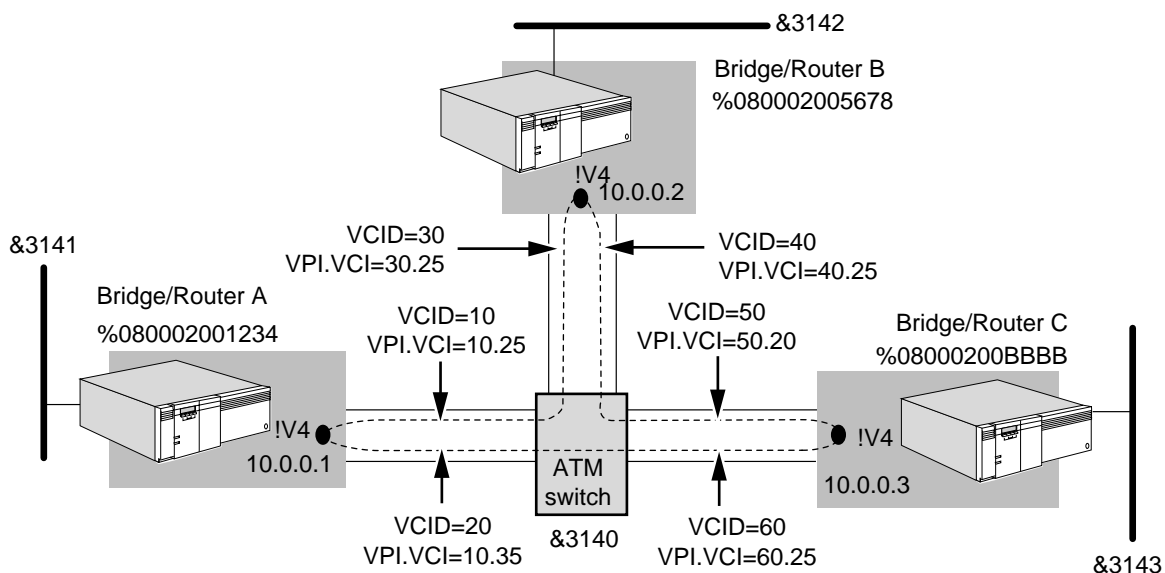
Before beginning this procedure, complete the following tasks:

- Configure your IPX LAN according to the procedures in the Configuring IPX Routing chapter.
- Set up the ATM Service as described in "Setting Up the ATM Service" earlier in this chapter.
- Determine the IPX network number to be assigned to each port attached to the ATM network.
- Obtain the MAC addresses for each remote host participating in IPX routing.

Procedure

To enable IPX to operate over an ATM switch, see Figure 418 and follow these steps:

Figure 418 Configuring IPX over ATM



- 1 Assign a network number to each virtual port on each 3Com bridge/router connected to the ATM switch.

For example, assign &3140 as the network number to virtual port 4 on bridge/routers A, B, and C by entering the following command on each bridge/router:

```
SETDefault !V4 -IPX NETnumber = &3140
```

- 2 Specify IPX network number to ATM VCID mapping information for each bridge/router directly connected to the ATM switch.

For example, to map bridge/router A's local VCID to the neighbor's MAC address, enter:

```
ADD !V4 -IPX ADDRESS &10 %080002005678
```

```
ADD !V4 -IPX ADDRESS &20 %08000200BBBB
```

Enter similar commands on bridge/routers B and C using their local VCIDs and the neighbor's MAC address. To obtain the physical MAC address of neighbors, enter:

SHow -SYS ADDRESS

- 3 If you have a partially meshed topology and you are operating on a non-NBMA network, specify which neighbors on each interface receive route reachability information.

For example, on bridge/router A, specify that bridge/router B receives route reachability information by entering:

```
ADD !V4 -NRIP AdvToNeighbor &3140%080002005678
ADD !V4 -SAP AdvToNeighbor &3140%080002005678
```



The dynamic neighbor learning feature is the default on ports on NBMA networks, such as X.25 and Frame Relay. This option is not displayed for non-NBMA networks. When dynamic neighbor learning is enabled, the neighbor list is automatically created and NRIP/SAP operates correctly without requiring you to manually configure static neighbor information as shown in the example in this step.

- 4 Enable the use of policy parameters by entering:

```
SETDefault !V4 -NRIP PolicyControl = AdvToNbr
SETDefault !V4 -SAP PolicyControl = AdvToNbr
```

- 5 Verify that IPX routing is enabled on each bridge/router that is attached to the ATM switch by entering:

SHow -IPX CONFiguration

If routing has been disabled on bridge/router A, enable it by entering:

```
SETDefault !V4 -IPX CONTROL = RRoute
```

Enable routing on bridge/routers B and C.

In this example, bridge/routers A, B, and C are running software version 9.0 or higher.

- 6 If you are using NRIP and SAP as your routing protocols, verify that routing is enabled on each port of each bridge/router that is attached to the ATM switch by entering:

SHow -NRIP CONTROL

To verify that Auto, or Talk and Listen are set, enter:

SHow -SAP CONTROL

- 7 If you are using NLSP as the routing protocol, follow these steps:

- a Make sure the NLSP routing protocol is enabled by entering:

SHow -NLSP CONTROL

- b Specify the local VCID of the PVC that is associated with neighbors that will be taking part in routing over ATM using:

```
ADD !<port> -NLSP Neighbors &<VCID>
```

For example, on bridge/router A enter the local VCIDs of the PVCs:

```
ADD !V4 -NLSP Neighbors &10
ADD !V4 -NLSP Neighbors &20
```

To allow the bridge/routers B and C to accept the adjacency, you must configure the Neighbors parameter on each of them and supply the local VCID.

- c Display the NLSP adjacencies by entering:

SHow -NLSP ADJAcencies

This completes the procedure for configuring IPX routing over an ATM switch.

How ATM Works

ATM transmits voice, video, and data across LANs, MANs, and WANs. ATM is an international standard defined by the American National Standards Institute (ANSI) and the International Telecommunications Union–Telecommunications Standards Sector (ITU-TSS), formerly CCITT. ATM is the result of research and the development of the Broadband Integrated Services Digital Network (B-ISDN).

ATM implements a high-speed, connection-oriented, cell-switching and multiplexing technology provides you with bandwidth up to 155 Mbps (3Com's offering). In ATM, all information is formatted into small, fixed-length units called *cells*. Each cell contains 53 octets divided into a 48-octet information field (or payload) and a 5-octet header. By using small fixed-length cells with switching technology, ATM can provide minimal delays for voice and video applications. The switch processes each cell more quickly, and the switch throughput increases. Small cells are not delayed by large cells because all the cells are the same size, which greatly reduces network delays.

ATM operates in a connection-oriented mode. A connection-oriented service requires that a virtual connection be established between the source and destination nodes before data can be transmitted. All connections are virtual in the sense that bandwidth is not permanently assigned to the connection; instead, the network provides the required bandwidth when cells are transmitted. Connections can be established at subscription time as PVCs or on demand as switched virtual circuits (SVCs) using a signaling protocol. Only PVCs are supported.

Network Interfaces

Software versions 9.0 and higher support the ATM Forum's ATM User-Network Specification, version 3.0 and 3.1. In this specification, two types of interfaces are defined for ATM networks and are shown in Figure 419:

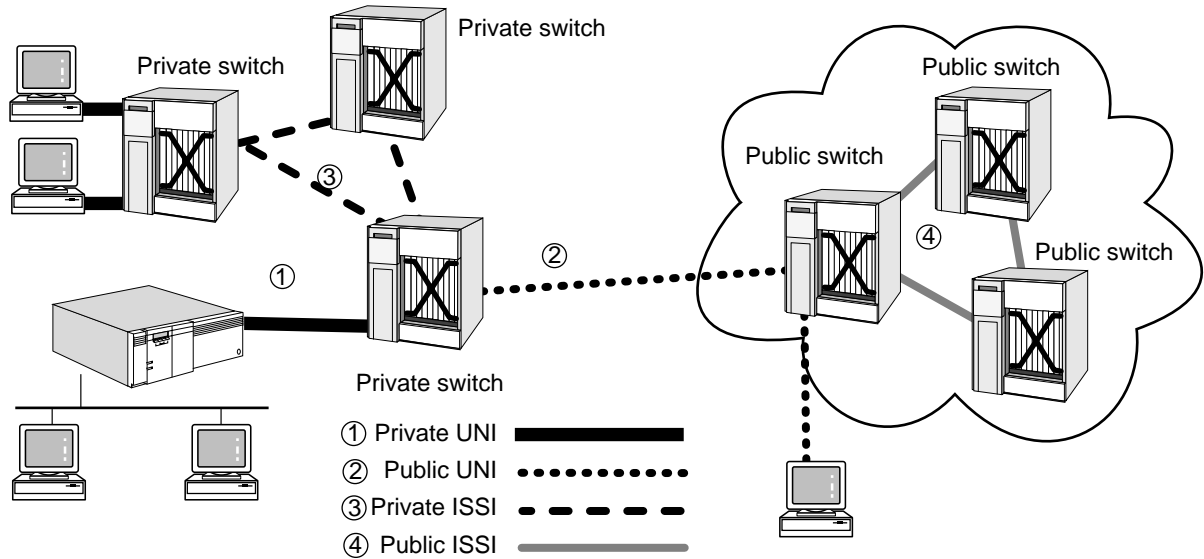
- User-to-network interface (UNI)
- Network node-to-network node interface (NNI)

The UNI defines the interface between a user and the network, and includes both private and public interfaces. In Figure 419, the private UNI (1) defines the interface between an ATM user device and a private ATM switch owned by a private organization. The public UNI (2) defines the interface between an ATM user device or a private ATM switch and an ATM switch used in a public service provider's network.

The NNI defines a switch-to-switch interface, also known as an inter-switching system interface (ISSI), and includes both private and public interfaces. In Figure 419, a private ISSI (3) defines an interface between private ATM switches. A public ISSI (4) defines an interface between public switches. The NNI does not

include the interface between a private switch and a public switch, which is considered part of the public UNI.

Figure 419 ATM Network Interfaces

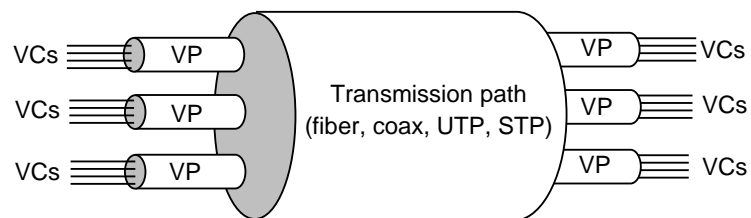


ATM Addressing, Virtual Paths, and Virtual Channels

The header of each ATM cell contains addressing information like traditional LAN packets. Instead of a specific destination address, each cell contains two fields, an 8-bit VPI and a 16-bit VCI, that specify the PVC over which the cell should be forwarded. The VPI and VCI fields define a routing field that provides an ATM switch with the information that it needs to route each cell. The PVC is usually represented in VPI.VCI format, where VPI is a decimal number between 0 and 255 and VCI is a decimal number between 0 and 65,535.

A virtual channel (VC) is a communications circuit that transports ATM cells between two or more endpoints. The endpoints of a VC can be a user-to-user connection, a user-to-network connection, or a network-to-network connection. When multiple VCs on the same transmission path are headed for the same destination, they can be grouped into a virtual path, which is a collection of VCs. A VP performs the same functions as a trunk line in a telephone network; the VP allows a number of virtual channels to be bundled together for transport between two ATM devices. Figure 420 shows the relationship between virtual channels and virtual paths.

Figure 420 Virtual Channels and Virtual Paths



When configuring your bridge/router for ATM, to configure a PVC use:

```
ADD !<port> -ATM PermVirCircuit <vcid> <vpi.vci> [LLCSNAP | [NULL | IP | IPX]] [<shaper_id>]
```

Encapsulation Types

Multiprotocol encapsulation over ATM AAL5 (MPATM) is supported using PVCs as defined in RFC 1483. The following encapsulation formats are supported for transparent bridging, and IP and IPX routing:

- MPATM logical link control/Subnetwork Access Protocol (LLC/SNAP)
Use LLC/SNAP encapsulation to allow multiple protocol types to be carried within a single ATM connection (virtual circuit). The type of the encapsulated packet is indicated by a standard LLC/SNAP header.
- NULL encapsulation
Use NULL encapsulation when only one protocol is configured to run on a VC. In this situation, no encapsulation is required. This type of encapsulation is not supported with transparent bridging.

For detailed descriptions of the encapsulation formats, see RFC 1483.

Quality of Service

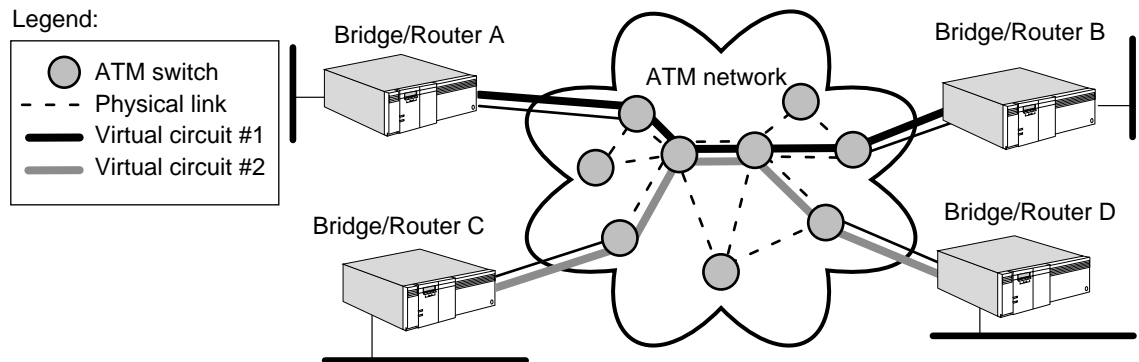
Different types of applications require different levels of service from a network. For example, voice and video applications are very sensitive to delay and variations in delay, but are not insensitive to minimal cell loss. Data applications are not insensitive to delay or variation in delay, but extremely sensitive to cell loss.

To meet the specific service requirements of each application, the node requesting the connection informs the network about the desired characteristics of each connection request. Some of the information in a connection request includes the following:

- Called party number
- Average bandwidth requirements
- Peak bandwidth requirements
- Maximum acceptable percentage of cell loss
- Maximum acceptable variation in network delay

The network uses this information to select the individual physical links that support the virtual circuit across the network as shown in Figure 421. For example, when selecting a specific physical link, the network makes sure that it can support all virtual circuits assigned to the physical link and still maintain the quality of service requirements for each individual virtual circuit. When the network and user agree on the characteristics of the connection, the network establishes the virtual circuit across the network. If the network cannot support the desired quality of service for a connection request, it rejects the connection.

Figure 421 Physical Links and Virtual Circuits



After the connection is established, the nodes at each end of the connection exchange information by transmitting cells across the UNI. The cells are relayed from switch to switch until they arrive at the UNI of the destination node. When there is no more data to be transmitted, the connection is terminated and the previously allocated network resources can be used by other connections.

- Traffic Shapers** A traffic shaper defines the attributes that allow the outbound traffic of attached virtual circuits to be transmitted based on the following items:
- Priority level
 - Average and peak rate in kilobits per second
 - Burst count

The peak rate specifies the maximum data rate at which a virtual circuit can transmit, which determines the maximum bandwidth available to all of the virtual circuits attached to the traffic shaper. You configure traffic-shaping attributes using the `-ATM TrafficShaper` parameter. You must associate every virtual circuit with one traffic shaper using the `-ATM PermVirCircuit` parameter.

A traffic shaper activates only when one or more of the attached virtual circuit connections becomes active. Each active traffic shaper consumes a fixed portion of the total bandwidth available on the associated ATM interface, as specified by the peak rate, regardless of the number of VCs that are attached to the traffic shaper.

The combined peak rates of all active traffic shapers should not exceed the maximum bandwidth available on the ATM interface. If the maximum bandwidth is exceeded, some traffic shapers and the associated virtual circuit traffic are not serviced because of the limitation in available bandwidth. For example, suppose you configure three active traffic shapers with the same priority and a peak rate of 75 Mbps and all of the attached virtual circuits are transmitting. The VCs attached to one of the traffic shapers will not be adequately serviced because the traffic shaper is selected one at a time in a round-robin process until the maximum bandwidth of 155 Mbps is reached.

The software displays the following message if the total peak rate for all active shapers exceeds the maximum bandwidth:

```
WARNING: ATM traffic shapers configured for !<path> exceeds 155Mbps.
```

The software services traffic shapers configured with a high priority ahead of the shaper with low priority. If VCs attached to the high-priority shapers use up the available bandwidth, the VCs associated with the low-priority shapers are not serviced.

The software provides 14 traffic shapers with predefined initial values. You can reconfigure each traffic shaper to meet the traffic control requirements of the attached virtual circuits. The new traffic-shaping attributes do not take effect for the attached VCs until the associated ATM interface is reset (the path must be re-enabled). To display the predefined initial values of the traffic-shaping attributes, enter:

```
SHow -ATM TrafficShaper
```

Each of the 14 traffic shapers has a peak bit rate, average bit rate, burst cell rate, and a priority. Each virtual channel connection (VCC) present on the module must be mapped to a shaper for it to effectively carry data. When more than one VCC is mapped to a shaper, each VCC has the bandwidth defined by the shaper. The aggregate bandwidth of all the VCCs mapped to all the active shapers should not exceed the total bandwidth of the link. Shapers available on the ATMLink module provide the following features:

- Outbound data traffic control
- Bandwidth reservation
- Prioritization of traffic among VCCs of the same or different protocols

Examples of these features are shown in the following pages. For all examples, a maximum bandwidth of 50 Mbps full duplex is assumed.

Outbound Data Traffic Control

Where data is known to be of a variable rate and bursty in nature, the traffic shapers moderate and limit the traffic rate to a predefined shaper value. The following example illustrates outbound data traffic control.

Example To limit IP traffic going from router A to router B to a peak rate of 15 Mbps, an average rate of 10 Mbps, and a maximum number of back-to-back cells at the peak rate to 32 cells, follow these steps:

- 1 Define the shaper by entering:

```
SETDefault -ATM TrafficShaper = 3 15 10 32 H
```

- 2 Define the PVC and map it to the shaper:

```
ADD !V1 -ATM PermVirCircuit 1 10.20 null IP 3
```

Bandwidth Reservation

You can use bandwidth reservation where there are multiple protocols running and when bandwidth must be reserved for some protocols in a predetermined ratio.

Example Suppose IP and IPX protocols are running on the same UNI interface, and you want to reserve 35 Mbps for IP and 15 Mbps for IPX. Follow these steps:

- 1 Define a shaper for 30 Mbps average and peak rate by entering:

```
SETDefault -ATM TrafficShaper = 3 35 35 32 H
```

```
SETDefault -ATM TrafficShaper = 4 15 15 32 H
```

- 2 Define a second shaper for 25 Mbps average and peak rate by entering:

```
ADD !V1 -ATM PermVirCircuit 1 10.20 null IP 3
ADD !V1 -ATM PermVirCircuit 2 10.21 null IPX 4
```

Prioritization of Traffic among VCCs of the Same Protocol

When there are multiple VCCs for a given protocol, you can use prioritization between VCCs.

Example Suppose there are two VCCs defined to carry IP traffic, but you want the traffic on one VCC to be higher than the traffic on the other VCC. Follow these steps:

- 1 Define a set of traffic shaping attributes associated with each PVC by entering:

```
SETDefault -ATM TrafficShaper = 3 30 30 32 H
SETDefault -ATM TrafficShaper = 4 30 30 32 L
```

- 2 Add the PVCs on the virtual ports by entering:

```
ADD !V1 -ATM PermVirCircuit 1 10.20 null IP 3
ADD !V2 -ATM PermVirCircuit 2 10.21 null IP 4
```

Prioritization of Traffic among VCCs of Different Protocols

When there are multiple protocols, you can prioritize one protocol over the other.

Example Suppose there are two VCCs defined to carry IP and IPX traffic, but you want IP traffic to be a higher priority than IPX. Follow these steps:

- 1 Define a set of traffic-shaping attributes associated with each PVC by entering:

```
SETDefault -ATM TrafficShaper = 3 30 30 32 H
SETDefault -ATM TrafficShaper = 4 30 30 32 L
```

- 2 Add the PVCs on the virtual ports by entering:

```
ADD !V1 -ATM pvc 1 10.20 null IP 3
ADD !V2 -ATM pvc 2 10.21 null IPX 4
```

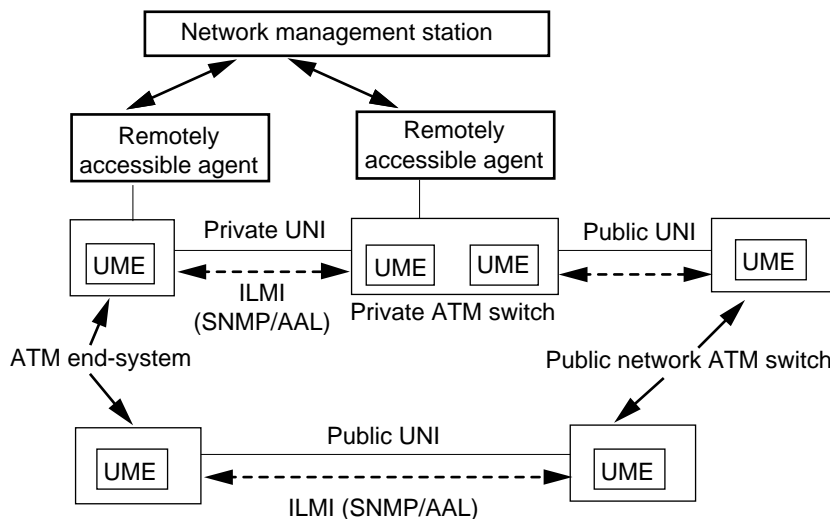
Network Management

When you connect to an ATM network using an ATM adapter on the user side to an ATM switch on the network side, your user-to-network connection is managed by an ATM UNI Management Entity (UME).

The UMEs exist on both sides of the interface and support an exchange of management information between them. UMEs are used in any device that transmits data in ATM cells across an ATM public or private UNI as shown in

Figure 422. Typical devices containing UMEs include workstations, bridges, routers, Frame Relay switches, and ATM network switches.

Figure 422 Interim Local Management Interface Definition



The two UMEs (one on each side of the UNI) have the same management information base (MIB) defined as the ATM UNI Interim Local Management Interface (ILMI) MIB by the UNI specification, and support seven groups of management information with respect to the user-to-network interface.

UMEs communicate using the ILMI Protocol, which uses SNMP version 1 PDUs encapsulated in AAL5. The ILMI provides status, configuration, and control information about the virtual path and virtual channel connections available at the UNI. You can obtain statistics about the status and operation of the UNI to facilitate performance monitoring and troubleshooting. By default, all ILMI communication takes place over the VCC with VPI = 0 and VCI = 16.

The key functions of the UME in the bridge/router software are as follows:

- Provides the SNMP agent on the NETBuilder II bridge/router access to all supported objects on the ATM UNI ILMI MIB groups (except for the network prefix group).
Access by the agent to the ATM UNI ILMI MIB on the switch is not supported. Access to other MIBs on the NETBuilder II bridge/router through the ILMI from the switch is also not supported.
- Provides a management station on the switch side access to all objects of the ATM UNI ILMI MIB as well as the "system" group.

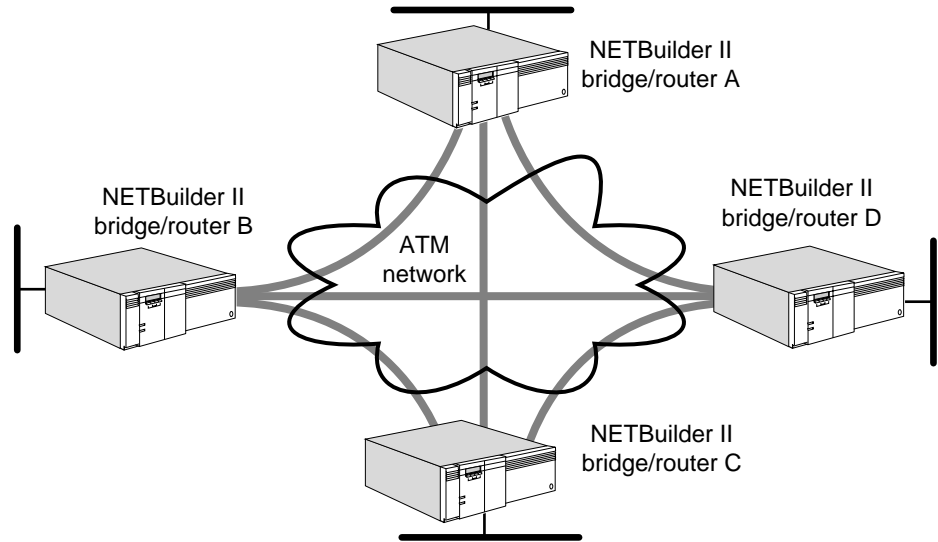
Fully Meshed, Partially Meshed, and Nonmeshed Topologies

A fully meshed ATM topology (Figure 423) is a topology in which each node on a network is directly connected to all other nodes on the network. Each node is connected to the other nodes through a virtual circuit.

The topology in Figure 423 consists of NETBuilder II bridge/routers. Using virtual circuits, bridge/router A is connected to bridge/routers B, C, and D; bridge/router B is connected to bridge/routers A, C, and D; and so on. This type of topology can

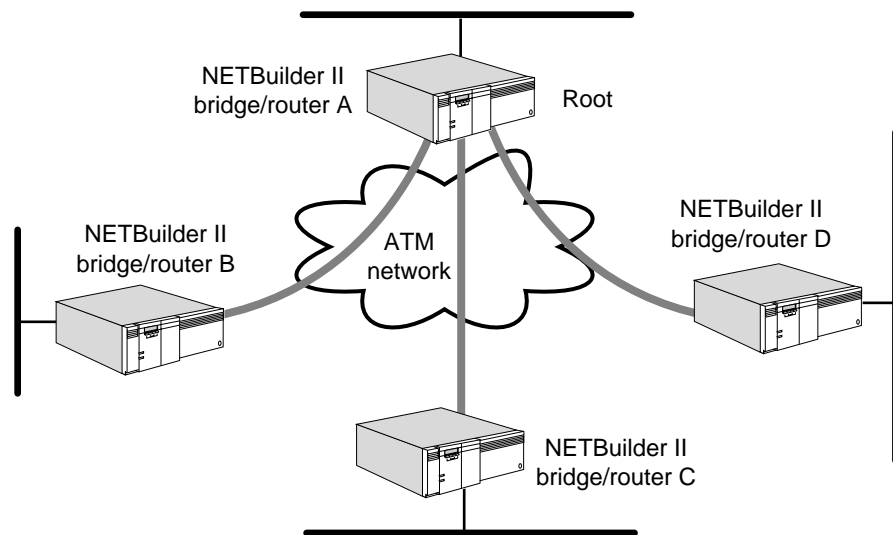
provide basic connectivity for campus backbones at 155 Mbps and also can construct sophisticated router clusters around one or more ATM switches.

Figure 423 Fully Meshed ATM Topology



A nonmeshed ATM topology (Figure 424) is a topology where each node on a network may not be connected to all other nodes on the network.

Figure 424 Nonmeshed ATM Topology



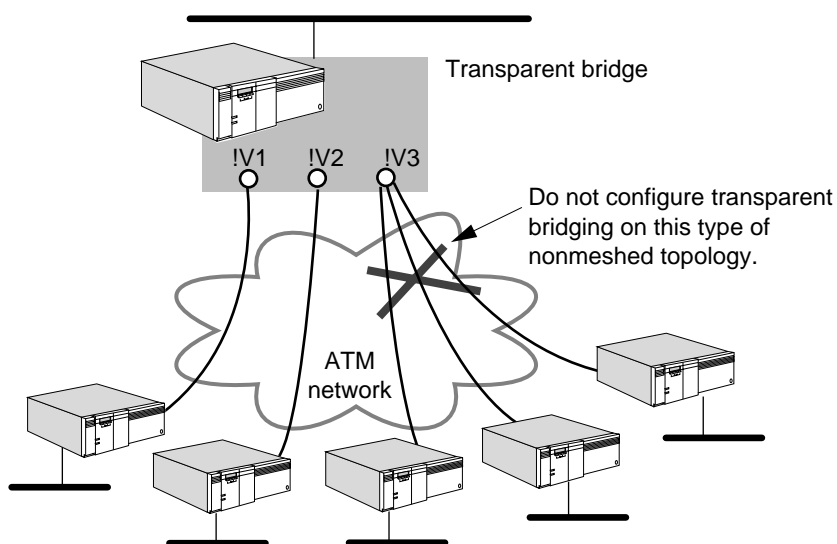
The topology in Figure 424 consists of NETBuilder II bridge/routers. Through the established PVCs, bridge/router A is connected to bridge/routers B, C, and D. bridge/routers B, C, and D are connected to bridge/router A only, but not to one another.

Nonmeshed topologies are supported but not recommended for use with ATM. Because each router is not connected to all other routers, traffic may have to cross the ATM switch twice. In Figure 424, traffic from bridge/router B to bridge/router

C must pass through the ATM switch to bridge/router A, which sends the traffic through the ATM switch again to bridge/router C. Because the traffic passes through the switch twice, the nonmeshed topology reduces the effectiveness of a high-speed ATM campus backbone.

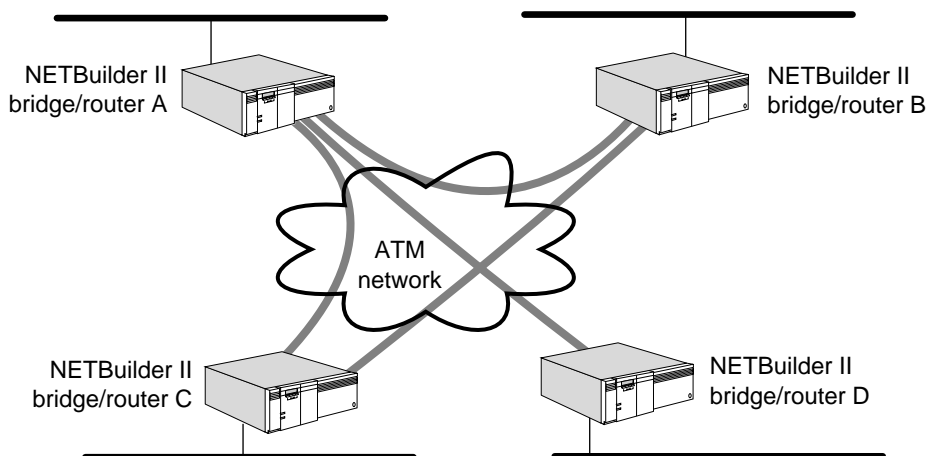
Transparent bridging does not correctly operate in some nonmeshed topologies. For example, in Figure 425, the transparent bridge properly forwards traffic received on !v1 to !v2. However, traffic received from one of its remote connections on !v3 is not properly forwarded to the other two remote connections on !v3; therefore, do not configure transparent bridging in this type of nonmeshed topology. The flooding algorithm floods packets on a per-port basis, not on a neighbor-per-port basis.

Figure 425 Transparent Bridging in Nonmeshed ATM Topologies



A partially meshed ATM topology is a topology where some nodes on a network are directly connected to nodes on the network (as in a fully meshed topology) and other nodes are not directly connected (as in a nonmeshed topology). Figure 426 is an example of a partially meshed ATM topology.

Figure 426 Partially Meshed ATM Topology



The topology in Figure 426 consists of four NETBuilder II bridge/routers. Through the established PVCs, bridge/routers A, B, and C are connected to one another but bridge/router D is connected to bridge/router A only.

The lack of connectivity among bridge/routers B, C, and D in partially meshed and nonmeshed topologies can be worked around using next-hop split horizon and virtual ports. If you are routing IP-RIP or IPX, these protocols offer the next-hop split horizon feature. In IP-RIP, set -RIP CONTROL to NonMesh to enable next-hop split horizon. In IPX, next-hop split horizon is enabled by manually configuring neighbors.

For example, if you are routing IP-RIP and you set -RIP CONTROL to NonMesh, a list of neighbors containing bridge/routers B, C, and D will be generated by the system, or you can configure them as neighbors using the -RIP AdvToNeighbor parameter. For more information about these parameters, see the RIP Service Parameters chapter in *Reference for Enterprise OS Software*.

If you are routing IPX, you can configure bridge/routers B, C, and D as neighbors using the -NRIP PolicyControl and -NRIP AdvToNeighbor parameters. For more information on next-hop split horizon, see the Configuring IP Routing chapter and the Configuring IPX Routing chapter.

Virtual ports are supported by bridging and all routing protocols, and must be used when configuring ATM for fully meshed, partially meshed, and nonmeshed topologies. For information on the number of virtual ports supported per platform, see Table 11 in the Configuring Advanced Ports and Paths chapter.

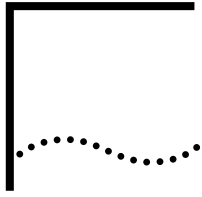
ATM Terms

The following terms are used in this chapter to explain ATM:

| | |
|---|--|
| Asynchronous Transfer Mode (ATM) | A transmission protocol that segments user traffic into small, fixed sized cells. Cells are transmitted to their destination where the original traffic is reassembled. |
| ATM Adaptation Layer (AAL) | Layer 3 of the ATM architecture that adapts user traffic into or from ATM 48-byte payloads.

AAL5 supports variable bit rate, delay tolerant, connection-oriented data traffic requiring minimal sequencing or error detection support. |
| Interim Local Management Interface (ILMI) | Refers to ATM Forum-defined interim specifications for network management functions between an end user and a public or private network, and between a public network and a private network. It is based on a limited subset of SNMP capabilities. |
| permanent virtual circuit (PVC) | A virtual channel connection that has been established by manual or semi-automated methods. It is similar to a leased or dedicated real circuit. |
| switched virtual circuit (SVC) | A virtual channel connection that has been dynamically established in response to a signaling request message. |
| UNI Management Entity (UME) | The code residing in ATM devices at each end of a UNI circuit that implements the management interface to the ATM network. |

| | |
|-----------------------------------|--|
| user-to-network interface (UNI) | ATM Forum-developed specifications for the procedures and protocols between a user DTE and the ATM network to effectively use ATM services and capabilities. |
| virtual channel connection (VCC) | Virtual channels in two or more sequential physical circuits concatenated to create an end-to-end connection. A VCC is a specific instance of a switched virtual circuit (SVC) or a permanent virtual circuit (PVC). |
| virtual channel identifier (VCI) | The 16-bit number in an ATM cell header identifying the specific virtual channel on which the cell is traversing on the current physical circuit. |
| virtual circuit identifier (VCID) | A user-assigned identifier or alias for a PVC representing the circuit characteristics. The VPI.VCI is analogous to the DLCI of a Frame Relay PVC. |
| virtual path identifier (VPI) | The 8-bit number in an ATM UNI cell header identifying the specific virtual path on which the cell is traversing on the current physical circuit. |



CONFIGURING INTERNETWORKING USING ATM AND LAN EMULATION

This chapter describes how to configure a NETBuilder II bridge/router to establish LAN, WAN, and MAN connectivity through Asynchronous Transfer Mode (ATM) with LAN emulation.



For conceptual information, see "How ATM and LAN Emulation Work" later in this chapter.

Setting Up the ATMLE Service

This section describes how to configure your bridge/router to transmit and receive data over an ATM interface using LAN emulation. Two procedures are provided: one for setting up an Ethernet configuration and one for setting up a Token Ring configuration with transparent bridging. See "Setting Up LAN Emulation Client Source Routing" later in this chapter for a procedure for setting up a Token Ring network for LAN emulation client source routing.

For detailed descriptions of all commands, see *Reference for Enterprise OS Software*.

Prerequisites

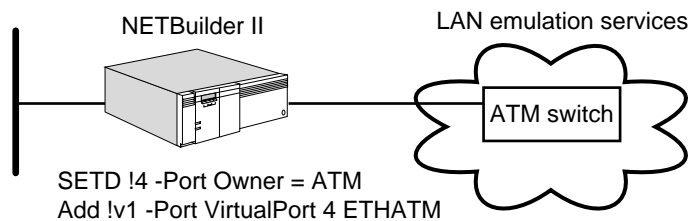
Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Make sure the ATM switch is connected and signalling is present on the interface.
- Note the path number of the ATM interface you wish to configure.
- Check the configuration of the LAN Emulation Server (LES) and determine the name(s) of the type(s) of Emulated LANs you want to use.

Procedure for Ethernet LANE

To perform LAN emulation over an ATM network, see Figure 427 and follow these steps:

Figure 427 Enabling a Port for LAN Emulation on Ethernet



- 1 Specify the emulated LAN name using:

```
SETDefault !<vport> -ATMLE ElanName = "<string>"
```

For example, to configure virtual port 1 to use the name "elan1", enter:

```
SETDefault !v1 -ATMLE ElanName = "elan1"
```

- 2 Create a virtual port for a LAN emulation client (LEC) to be attached to the ATM network using:

```
ADD !<port> -PORT VirtualPort <path> ETHATM
```

For example, to configure a virtual port (Ethernet) for path 4 on bridge/router A, enter:

```
ADD !v1 -PORT VirtualPort 4 ETHATM
```

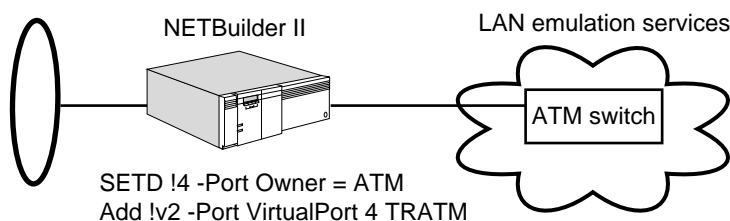
Each ATM LAN emulation virtual port has a unique MAC address, and virtual port numbers must be in the range !v1 through !v256. A total of 32 ATM LAN Emulation virtual ports may be created.

When the virtual port is added to the configuration, the ATM address for the virtual port is constructed using the MAC address as the ATM address end-system identifier.

Procedure for Token Ring LANE

To perform Token Ring LAN emulation over an ATM network, see Figure 428 and follow these steps:

Figure 428 Enabling a Port for LAN Emulation on Token Ring



- 1 Specify the emulated LAN name using:

```
SETDefault !<vport> -ATMLE ElanName = "<string>"
```

For example, to configure virtual port 2 to use the name "elan2", enter:

```
SETDefault !v2 -ATMLE ElanName = "elan2"
```

- 2 Create a virtual port for a LAN emulation client (LEC) to be attached to the ATM network using:

```
ADD !<port> -PORT VirtualPort <path> TRATM
```

For example, to configure a virtual port (Token Ring) for path 4 on bridge/router A, enter:

```
ADD !v2 -PORT VirtualPort 4 TRATM
```

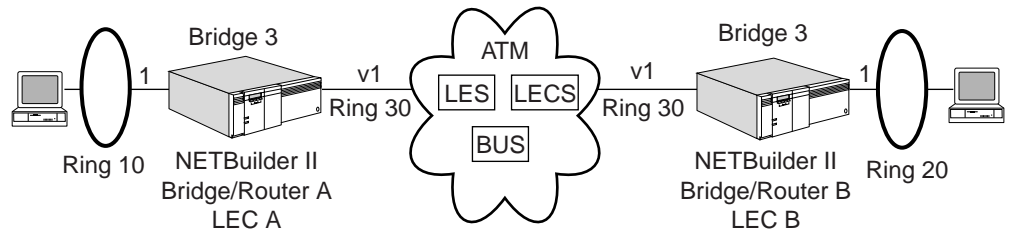


When only transparent bridging is being used, the source route ring number may be omitted.

Setting Up LAN Emulation Client Source Routing

To perform Token Ring LAN Emulation client source routing over an ATM network, see Figure 429 and follow these steps:

Figure 429 LAN Emulation Client Source Routing



The following procedure shows how to set up a source routed LEC where both NETBuilder bridge/routers join the same emulated source routed LAN. Notice that both source routed LEC ports must be configured with the same ring number.

On NETBuilder bridge/router A, follow these steps:

- 1 Specify the emulated LAN name by entering:

```
SETDefault !v1 -ATMLE ElanName = "elan3"
```

- 2 Create a virtual port for a Token Ring LEC to be attached to the ATM network using:

```
ADD !<vport> -PORT VirtualPort <path> TRATM
```

For example, to configure virtual port 3 (token ring) for path 1 on bridge/router A, enter:

```
ADD !v1 -PORT VirtualPort 1 TRATM
```

- 3 Establish the source route ring number 30, using:

```
SETDefault <!vport> -SR RingNumber=<1...4095>
```

For example, enter:

```
SETDefault !v1 -SR rn=30
```

- 4 Configure the ring number for the Token Ring port, the bridge number, and turn on the bridge by entering:

```
SETDefault !1 -SR RingNumber=10
```

```
SETDefault -SR BridgeNumber=3
```

```
SETDefault -BRIDGE CONTROL=Bridge
```

On NETBuilder bridge/router B, follow these steps:

- 1 Specify the emulated LAN name by entering:

```
SETDefault !v1 -ATMLE ElanName = "elan3"
```

- 2 Create a virtual port for a source routed LEC to be attached to the ATM network using:

```
ADD !<vport> -PORT VirtualPort <path> TRATM
```

For example, to configure virtual port 1 (Token Ring) for path 1 on bridge/router B, enter:

```
ADD !v1 -PORT VirtualPort 1 TRATM
```

- 3 Establish the source route ring number 30, using:

```
SETDefault <!vport> -SR RingNumber=<1...4095>
```

For example, enter:

```
SETDefault !v1 -SR rn=30
```

- 4 Configure the ring number for the Token Ring port, the bridge number, and turn on the bridge by entering:

```
SETDefault !v1 -SR RingNumber=20
```

```
SETDefault -SR BridgeNumber=3
```

```
SETDefault -BRIDGE CONTROL=Bridge
```

Verifying the Configuration

To verify your ATM LAN emulation configuration, display current ATM configuration information use:

```
SHowDefault !<port> -ATMLE CONFIGuration
```

Verify that your ATM LAN emulation configuration parameters are configured correctly.

Controlling Initialization

During initialization the LEC can either rely on the ATM switch unit management entity (UME) to determine the ATM address of the LEC's LAN emulation configuration server or configure the ATM address of a specific LECS.

The LECSATMAddr parameter specifies the ATM address of the LECS. When the LEC is in "manual" mode, and the LECSATMAddr parameter is configured, the LEC uses the configured ATM address to connect to the specified LECS. When the LEC is in "automatic" mode, it uses the UME to retrieve the LECS ATM address that will be used during initialization.

To specify which LECS address to use during initialization, follow these steps:

- 1 Specify the ATM address of the LECS to be used during initialization using:

```
SETDefault !<vport> -ATMLE LECSAddr <atm address>
```

For example, to assign the LECS with the ATM address 470079000000000000000000A03E000000100 as the LECS to be used during initialization, enter:

```
SETDefault !v4 -ATMLE LECSAddr 470079000000000000000000A03E000000100
```

- 2 Set the LEC to manual mode using:

```
SETDefault !<vport> -ATMLE CONTROL = ([ MANual | AUTOMATIC ], [ Proxy | NoProxy ])
```

For example, to enable manual mode on the LEC, enter:

```
SETDefault !v4 -ATMLE CONTROL = MANUAL
```

- 3 Enable the LEC virtual port by entering:

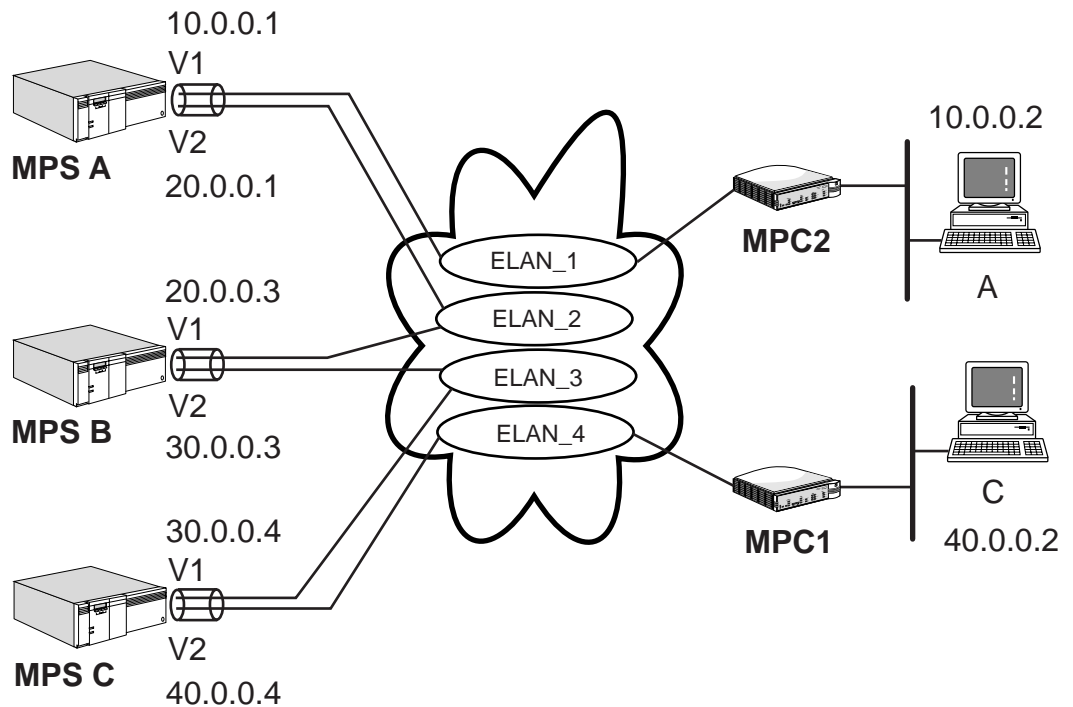
```
SETDefault !v4 -PORT CONTROL = ENabled
```

Configuring Multiprotocol Over ATM Services

Multiprotocol over ATM (MPOA) is a means to provide short-cut, inter-subnet virtual circuit connections in a LAN emulation topology. MPOA architecture consists of MPOA clients (MPCs), and MPOA servers (MPS).

Figure 430 is an example of an MPOA configuration.

Figure 430 Multiprotocol over ATM Configuration Example



Procedure To configure the MPS routers shown in Figure 430, follow these steps:

- 1 On the bridge/router MPS A, specify the emulated LAN names by entering:

```
SETDefault !v1 -ATMLE ElanName = "elan_1"
SETDefault !v2 -ATMLE ElanName = "elan_2"
```

- 2 Create virtual ports using:

```
ADD !<port> -PORT VirtualPort <path> ETHATM
```

For example, to configure the virtual port (Ethernet) for path 1 on bridge/router MPS A, enter:

```
ADD !v1 -PORT VirtualPort 1 ETHATM
ADD !v2 -PORT VirtualPort 1 ETHATM
```

- 3 Assign an IP address to each of the virtual ports on MPS A entering:

```
SETDefault !v1 -IP NETaddr = 10.0.0.1
SETDefault !v2 -IP NETaddr = 20.0.0.1
```

- 4 On the bridge/router MPS B, specify the emulated LAN names by entering:

```
SETDefault !v1 -ATMLE ElanName = "elan_2"
SETDefault !v2 -ATMLE ElanName = "elan_3"
```

- 5 Create virtual ports using:

```
ADD !<port> -PORT VirtualPort <path> ETHATM
```

For example, to configure the virtual port (Ethernet) for path 1 on bridge/router MPS B, enter:

```
ADD !v1 -PORT VirtualPort 1 ETHATM
ADD !v2 -PORT VirtualPort 1 ETHATM
```

- 6 Assign an IP address to each of the virtual ports on MPS B by entering:

```
SETDefault !v1 -IP NETaddr = 20.0.0.3
SETDefault !v2 -IP NETaddr = 30.0.0.3
```

- 7 On the bridge/router MPS C, specify the emulated LAN names by entering:

```
SETDefault !v1 -ATMLE ElanName = "elan_3"
SETDefault !v2 -ATMLE ElanName = "elan_4"
```

- 8 Create virtual ports using:

```
ADD !<port> -PORT VirtualPort <path> ETHATM
```

For example, to configure the virtual port (Ethernet) for path 1 on bridge/router MPS C, enter:

```
ADD !v1 -PORT VirtualPort 1 ETHATM
ADD !v2 -PORT VirtualPort 1 ETHATM
```

- 9 Assign an IP address for each of the virtual ports on MPS C by entering:

```
SETDefault !v1 -IP NETaddr = 30.0.0.4
SETDefault !v2 -IP NETaddr = 40.0.0.4
```

You may also want to adjust the operational setting for the Multiprotocol Over ATM Server using the MPS Service parameters. For more information, see the MPS Service Parameters chapter in *Reference for Enterprise OS Software*.

Normally, the routed path for packets exchanged between subnets must go through routers connecting the subnets. In Figure 430, station A and station C must go through the routers connecting ELAN_1, ELAN_2, ELAN_3, and ELAN_4. Since the edge devices providing station A and station C access to the ATM network are both physically attached to the same ATM network fabric, the edge devices should be able to connect directly with each other, therefore allowing station A and station C to bypass the intermediate routers in the data path. MPOA provides the capability for an edge device to resolve the ATM address of the edge device servicing a destination network protocol address. The edge devices connect to each other and bypass the intermediate routers.

In Figure 430, the packet enters the MPOA system at the incoming MPC (MPC1). MPC1 has already determined that the next hop router's MAC address belongs to a MPS (when the LE_ARP_RESPONSE packet resolved the router's MAC address to ATM address), MPS C. MPC1 creates a cache entry for the destination Internetworking address (e.g. IP address) and begins monitoring the flow to that destination. Once a flow is detected (number of packets sent to that destination exceeding some threshold), MPC1 puts together a MPOA Resolution Request for that destination and sends it on the MPOA VCC to MPS C.

When MPS C receives the MPOA Resolution Request, it examines the destination address specified in the MPOA Resolution Request. The destination address subnet is not a locally attached network. The next-hop towards the destination address is the router, MPS B. MPS A discovers that the MAC address associated with MPS B belongs to another MPS and re-originates the MPOA Resolution Request as a NHRP Resolution request. The packet is forwarded on the routed path through LANE Data Direct VCC to MPS B toward the destination. The re-originated NHRP request will have the MPS C's protocol address as the source protocol address and a new NHRP request ID derived from mapping the source ATM address, destination protocol address, and MPOA request ID.

MPS B receives the NHRP Resolution Request and determines that the next hop MAC address toward the destination specified in the request is another MPS (MPS

A) and forwards the request on the routed path through LANE Data Direct VCC to MPS A.

MPS A receives the NHRP Resolution Request and determines the destination address subnet is a locally attached network. MPS A inspects its MPOA cache and discovers that the destination protocol address next hop MAC address belongs to a MPC that it services. MPS A translates the NHRP Resolution Request to an MPOA Cache Imposition Request and sends it on the MPOA VCC to the outgoing MPC (MPC2).

MPC2 receives the MPOA Cache Imposition Request and creates a cache entry and responds to the Cache Imposition Request by returning an MPOA Cache Imposition Reply on the MPOA VCC to the outgoing MPS (MPS A).

The outgoing MPS (MPS A) then translates the MPOA Cache Imposition Reply to an NHRP Resolution Reply and forwards the reply through LANE Data Direct VCC on the routed path toward the incoming MPS (MPS C).

MPS B receives the NHRP Resolution Reply and forwards it through LANE Data Direct VCC toward the incoming MPS (MPS C). When the incoming MPS (MPS C) receives the NHRP Resolution Reply, it matches the ATM address, destination protocol address, request ID with an outstanding NHRP Resolution Request. MPS C translates the Reply to an MPOA Resolution Reply and sends it on the MPOA VCC to the incoming MPC (MPC1).

At the end of this process, the incoming MPC (MPC1) is prepared to establish a MPOA short-cut VCC and the outgoing MPC (MPC2) is prepared to receive data over the short-cut. MPC1 opens a VCC connection to MPC2 and associates the short-cut VCC connection to the IP address that had initiated the MPOA Resolution Request. When MPC1 detects a packet destined to that IP address, it will send the packet over the short-cut VCC connection.

How ATM and LAN Emulation Work

ATM transmits voice, video, and data across LANs, MANs, and WANs. ATM is an international standard defined by the American National Standards Institute (ANSI) and the International Telecommunications Union–Telecommunications Standards Sector (ITU-TSS), formerly CCITT. ATM is the result of research and the development of the Broadband Integrated Services Digital Network (B-ISDN).

ATM implements a high-speed, connection-oriented, cell-switching, and multiplexing technology that provides bandwidth up to 155 Mbps (NETBuilder's offering). In ATM, all information is formatted into small, fixed-length units called cells. Each cell contains 53 octets divided into a 48-octet information field (or payload) and a 5-octet header. By using small fixed-length cells with switching technology, ATM can provide minimal delays for voice and video applications. The switch processes each cell more quickly, and the switch throughput increases. Since packets are broken into small cells that are multiplexed on the ATM network backbone, smaller time-sensitive packets are less likely to be delayed by large packets on the network as often happens in a LAN environment.

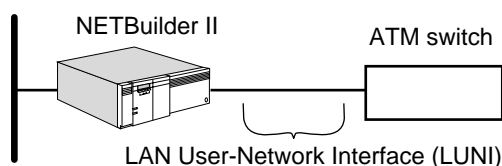
ATM operates in a connection-oriented mode. A connection-oriented service requires that a virtual connection be established between the source and destination nodes before data can be transmitted. All connections are virtual in the sense that bandwidth is not permanently assigned to the connection; instead,

the network provides the required bandwidth when cells are transmitted. Connections can be established at subscription time as permanent virtual circuits (PVCs) or on demand as switched virtual circuits (SVCs) using a signaling protocol.

Network Interfaces The Enterprise OS software supports the ATM Forum's ATM LAN Emulation User Network Specification version 1.0.

The interface for interoperability with legacy LANs and protocols is the LAN emulation user network interface (LUNI) shown in Figure 431. The LUNI protocols allow ATM-attached end systems and LAN/ATM conversion devices to control the virtual connections required for transmission and to emulate the connectionless nature of a LAN or LAN emulation.

Figure 431 LAN Emulation User Network Interface (LUNI)



The main objective of the LAN emulation specification is to enable existing applications to access an ATM network through protocol stacks such as APPN, NetBIOS, and IPX as if they were running over traditional LANs.

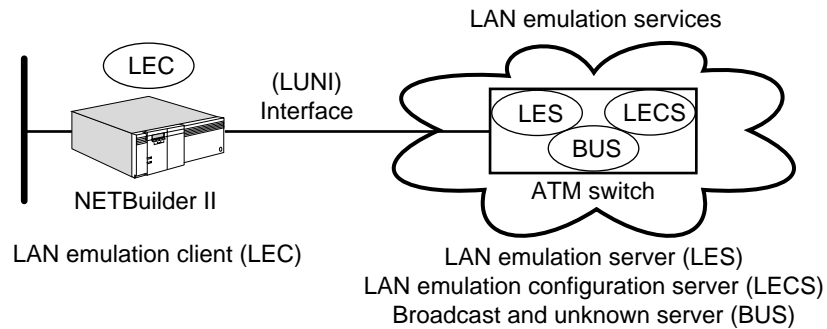
LAN emulation works at the media access control (MAC) layer and enables legacy Ethernet, token ring, or FDDI traffic to run over ATM with no modifications to applications network operating systems, or desktop adapters. Legacy end stations can use LAN emulation to connect to other legacy systems as well as to ATM-attached servers, routers, hubs, and other networking devices.

ATM Addressing The header of each ATM cell contains addressing information like traditional LAN packets. Instead of a specific destination address, each cell contains two fields, an 8-bit VPI and a 16-bit VCI, that specify the PVC or SVC over which the cell should be forwarded. The VPI and VCI fields define a routing field that provides an ATM switch with the information that it needs to route the cell. The PVC or SVC is usually represented in VPI.VCI format, where VPI is a decimal number between 0 and 255 and VCI is a decimal number between 0 and 65,535.

LAN Emulation LAN emulation is a method for carrying network layer packets across an ATM network. The function of the LAN emulation protocol is to emulate LAN while transporting the packets over an ATM network. The LAN emulation protocol defines the service interface for higher layer network protocols. This interface presents an identical appearance to the existing LANs, and data sent across the ATM network is encapsulated in appropriate LAN MAC packet format. The MAC protocol of the specific LAN is not emulated, whether the MAC protocol is either token passing for 802.5 network types or CSMA/CD for Ethernet types.

LUNI Components and Connections An emulated LAN on an ATM network consists of the elements shown in Figure 432.

Figure 432 LAN Emulation Entities



LAN Emulation Client

The LEC is a process in the Enterprise OS software that operates as an end system. The LEC forwards data, resolves addresses, and performs control functions for a single end-system. A LEC also provides a standard LAN service interface to any higher layer process that interfaces to the LEC.

Each LEC is identified by a unique ATM address, and is associated with one or more MAC addresses reachable through that ATM address.

LAN Emulation Configuration Server

The LECS is a process that assigns individual LAN emulation clients to particular emulated LANs by directing them to the LES that corresponds to the ELAN. There is logically one LECS per administrative domain, which serves all LAN emulation clients within that domain.

LAN Emulation Server

The LES provides the control functions for a particular emulated LAN. There is only one logical LES per emulated LAN, and to belong to a particular emulated LAN means to have a control relationship with that emulated LAN's particular LES. Each LES is identified by an ATM address. The LES ATM address is supplied to the LEC by the LECS or configured through the user interface.

Broadcast and Unknown Server

The Broadcast and Unknown Server (BUS) is a multicast server that is used to flood unknown destination address traffic and forward multicast and broadcast traffic to clients within a particular ELAN. Each LEC is associated with only a single BUS per ELAN, but there may be multiple BUSs within a particular ELAN. The BUS to which a LEC connects is identified by a unique ATM address. The BUS ATM address is supplied to the LEC by the LES.

Operation

The operation of a LAN emulation system consisting of the components described above consists of three main phases:

- Initialization and configuration
- Joining and registration
- Data transfer

Initialization and Configuration

When the interface becomes active, the LEC must get its ATM address. The LEC then sets up a configuration-direct connection to the LECS. The LEC must find the location of the LECS. The LECS address may be configured in the LEC and the

NETBuilder II bridge/router set to Manual so that the LEC sets up the configuration-direct connection with the specified LECS. The LEC also can rely on the UME of the ATM switch to determine an appropriate LECS address.

After finding the location of the LECS, the LEC establishes a configuration-direct VCC to the LECS. When successfully connected, the LECS uses a configuration protocol to inform the LEC of the information it requires to connect to its target ELAN. This information includes the ATM address of the LES, the type of LAN being emulated, the maximum packet size on the emulated LAN, and the emulated LAN name, which consists of a text string. Network management usually configures the LECS with this information.

Joining and Registration

When the LEC gets the LES address, it sets up the control-direct VCC to the LES. When this setup is complete, the LES assigns the LEC with a unique LEC Identifier (LECID). The LEC then registers its own MAC and ATM address with the LES.

The LES then sets the control distribute VCC back to the LEC by adding the LEC as a leaf to a point to multipoint connection. The control direct and distribute VCCs can then be used by the LEC for the LAN emulation ARP (LE_ARP) procedure for requesting the ATM address that corresponds to a particular MAC address. To do this, the LEC formulates an LE_ARP request and sends it to the LES. If the LES recognizes this mapping, it may choose to reply directly on the control-direct VCC. If it does not, it forwards the request on the control-distribute VCC to solicit a response from a LEC that knows the requested MAC address.

If a LEC can respond to the LE_ARP request because it is proxying for that address, the LEC responds to the LES on the control direct VCC. The LES then forwards this response either only to the requesting LEC, or, optionally, on the control-distributed VCC to all LECs. All LECs then can learn and cache the particular address mapping, preventing future LE_ARPs for that MAC address.

To complete registration, a LEC uses this LE_ARP mechanism to determine the ATM address of the BUS. The LEC determines the address by sending an LE_ARP for the MAC broadcast address to the LES, which responds with the BUS ATM address. The LEC then sets up the multicast-send VCC to the BUS. The BUS, then sets up the multicast forward VCC back to the LEC by adding the LEC as a leaf to a point-to-multipoint connection. The LEC is now ready to transfer data.

Data Transfer

When a LEC is ready to transmit a data frame onto an ELAN, it first checks its local tables to determine if the ATM address associated with the destination MAC address has already been learned. If it has not, the LEC sends the data frame to the BUS, which delivers a copy of the frame to every client on the ELAN

Simultaneously, the LEC sends an LE_ARP request to the LES, trying to resolve the unknown MAC address. The LE_ARP message includes the source ATM address of the LEC making the request. The LES searches its database of MAC address-to-ATM address mappings and returns the ATM address if known through an LE_ARP response. However, in most implementations the LES forwards the LE_ARP to all clients.

The target client recognizes the MAC address and sends an LE_ARP response to the LES, which includes both its own ATM address and the source ATM address for the LEC originating the LE_ARP request. The server forwards the response

message with the target ATM address to all the LECs in broadcast fashion. The cycle ends when the originating LEC recognizes its own ATM address contained in the response. At this point, it has learned the ATM address associated with the unknown MAC address and can set up a data-direct connection to the target LEC. When the LEC is ready to transmit subsequent data frames to the newly learned MAC address, the frames are forwarded on the associated data-direct VCC.

Each LEC builds up its own table of MAC addresses, ATM addresses, and VCC bindings. If a particular MAC address has not been active for some time. The LEC eventually drops it from its cache. When there are no more MAC addresses associated with a data-direct VCC, the connection will eventually be released due to inactivity.

Multiprotocol Over ATM Background

An MPOA configuration consists of MPSs, which are co-located with routers and next hop servers, and MPCs, which are co-located with LECs on MPOA hosts or edge devices.

In nonbroadcast, multiaccess networks such as ATM, all nodes are physically capable of communicating to each other via a direct virtual circuit connection (VCC). In the acceptance of ATM, networking customers using ATM for workgroup LANs and LAN backbones require coexistence with existing legacy LAN networks.

ATM Forum's LAN Emulation provided this migration path by allowing end stations (workstation and servers) to connect to the ATM network as though the end stations were connected to a LAN. LAN Emulation provides the ATM services that emulate the services of existing connectionless and multicast capable legacy LANs across a connection-oriented ATM network.

LAN emulation divided the ATM network into multiple Logical Internet Subnets (LISs) or Emulated LANs (ELANs), but required all inter-LIS traffic to go through routers that connected the LISs.

The penalty for this organizational convenience is that all traffic between the subnets must go through the router, rather than straight through the switch fabric. On a large site it is quite likely that there would be two or more routers on the data path between the end stations. If the two end stations are both physically attached to the same ATM network fabric, then the end stations should be able to communicate directly with each other, bypassing one or more intermediate routers in the data path.

The ATM Forum addressed this inefficiency by using the IETF RFC Next Hop Resolution Protocol (NHRP) to allow inter-subnet Internetworking Layer protocols to communicate over short-cut VCCs. NHRP allows intermediate routers connecting NBMA subnetworks to be bypassed on the data path by allowing the source station to resolve the NBMA address of the next hop toward the destination network protocol address.

NHRP consists of Next Hop Clients (NHCs) and Next Hop Servers (NHSs). The NHC (endstation) initiates the NHRP requests to the NHS (routing entity) to resolve the NBMA address of the NHC serving the destination network protocol address. The NHS contains the NBMA addresses of the stations (NHCs) that it serves through registration from the NHC or by configuration.

Using NHRP does not exactly fit in LANE. The NHRP protocol provides the means to find the NBMA, but it also requires a network protocol layer at the NHC (end points of the short-cut VCC) to deliver the packet to the source station. In LANE, the NHC functionality would reside in the LANE edge device (such as the CoreBuilder™ 7200 module in the CoreBuilder 7000 hub), but the LANE edge device is normally a bridge to a LAN and does not necessarily have an internetwork address or an internetworking layer protocol stack.

Since the MPC might not have a network protocol layer to resolve the destination protocol address to the data link layer address, additional MPOA specific control messages were used to augment the NHRP control messages. Using the MPOA control messages (specifically MPOA Cache Imposition Request and Reply), the MPC caches the data link layer information to allow the MPC to perform network layer forwarding, even though the MPC does not have a network protocol stack.

MPOA solves this problem by integrating LANE and NHRP to preserve the benefits of LAN Emulation and using NHRP to resolve the ATM address of the edge device servicing the destination network protocol address.

MPOA provides MPCs and MPSs and extends the NHRP packet type to include specific MPOA packet types for MPC and MPS communication. MPCs issue queries for ATM addresses and receive replies from the MPS using the MPOA packet types. Communication between MPS and MPS are done using the NHRP packet types.

The MPS are logically co-located with routers and performs the MPOA specific functionality (interacting with the MPC). To perform the NHRP functionality (forwarding of NHRP packets to the outgoing MPS), the MPS must have a NHS. The MPS make use of the standard internetworking protocols such as OSPF and RIP.

The MPC are normally co-located with an edge device that does not contain an internetworking layer protocol stack. To act as an internetwork forwarder when a network protocol packet is received on the shortcut VCC, the MPC contains an internetwork layer forwarding database that is administered by the MPS (containing the router).

The learning of a co-located MPC with an edge device or a co-located MPS with a router is automatically done through LANE because the LAN Emulation Clients (LECs) within the edge device and router already communicate with each other through LANE protocols. MPOA requires the extended TLVs defined in LANE Version 2.0 to allow the LECs to advertise their MPOA capability. Therefore LECs supporting LANE Version 1.0 in an MPOA architecture must support the LANE Version 2.0 features for MPOA.

Token Ring LAN Emulation Client

The NETBuilder II bridge/router provides IEEE 802.5 Token Ring LAN Emulation Client functionality. The Token Ring LEC includes all the functionality described above for IEEE 802.3 emulated LANs, as well as the source routing capabilities of IEEE 802.5 LANs. Token Ring LECs are capable of transparent bridging, source route bridging, and routing either with or without source route discovery. Since Token Ring LECs provide virtually identical operation to Ethernet LECs for transparent frames, only the additional functionality associated with source routing is discussed in this section.

- Source Routing** Source route LECs require information about the network topology to make forwarding decisions for source routed frames. This information takes the form of ring number, bridge number combinations, called Route Descriptors (RDs). A source route LEC uses RDs with the LE_ARP function to obtain the ATM address of the LEC attached to a particular ring in the source route path. These RD-to-ATM address mappings are stored by the LEC in a table similar to that used for MAC address mappings on IEEE 802.3 ELANs, and are used to setup data-direct VCCs to send traffic to the destination RD.
- RD Registration** To allow the LES to respond to LE_ARPs and provide RD mapping information to other LECs attached to the ELAN, a source route bridge LEC registers all of its RDs with the LES when joining the ELAN. The LEC constructs an RD for each source route ring behind the LEC using the ring number together with its bridge number. These RDs are delivered to the LES on the control-direct VCC using the LAN Emulation registration protocol.
- Data Transfer** There are three basic types of data frames used in source route networks. The first two, All Routes Explorer (ARE) and Spanning Tree Explorer (STE), are broadcast frames. These frames are sent to the BUS to be delivered to all the LECs on the ELAN. On a source route bridge LEC, ARE frames are always forwarded, while STE frames are only forwarded for ports that are in the forwarding state as determined by the spanning tree protocol.
- The third basic type of data frame used in source route networks is the Specifically Routed Frame (SRF). The source route LEC handles these frames in one of two ways based upon the information stored in the Routing Information Field (RIF) of the frame. If the source route LEC determines that the frame is destined to a station on the local ring (the attached ELAN), the frame is forwarded based upon the MAC address-to-ATM address mapping in the same way as on IEEE 802.3 ELANs (described earlier).
- If the SRF is destined to a station on a ring beyond the local ring, the LEC uses the information in the frame's RIF to construct an RD consisting of the next-hop bridge number and ring number. The LEC then checks to see if this RD has already been entered in its local RD table. If it has not, the LEC sends an LE_ARP request to the LES for the frame's RD and queues the frame until a data-direct VCC is established (a maximum of 20 frames may be queued).
- When the LES receives the LE_ARP request for the RD, it looks in its table of registered RDs to find the ATM address of the LEC that registered the specific RD. The LES gives this ATM address to the LEC in an LE_ARP reply message. The LEC then saves this RD-to-ATM address mapping in its local RD table. If the LEC does not already have a data-direct VCC to the ATM address associated with the RD, it sets up the connection. At this point any frames that have been queued up for this data-direct VCC are transmitted, and as long as the RD is active, all subsequent SRFs destined to the RD are forwarded on this connection.
- If a particular RD is not used for a period of time, it will be dropped from the RD table. When all the RDs associated with a particular data-direct VCC have been removed from the RD table, the connection is released.

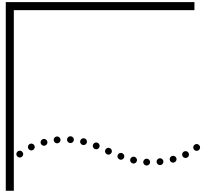
ATM LAN Emulation Terms

The following terms are used in this chapter to explain ATM:

| | |
|---|--|
| ARE packets | All Route Explorer (ARE) packets used for Route Discovery or traversing to all possible routes between end stations in source route bridge environment. |
| Asynchronous Transfer Mode (ATM) | A transmission protocol that segments user traffic into small, fixed-sized cells. Cells are transmitted to their destination where the original traffic is reassembled. |
| ATM Adaptation Layer (AAL) | Layer 3 of the ATM architecture that adapts user traffic into or from ATM 48-byte payloads.

AAL5 supports variable bit rate, delay-tolerant, connection-oriented data traffic requiring minimal sequencing or error-detection support. |
| ATMUME | ATM UNI management entity component that provides the interfaces for the LEC to receive status and transmit/receive frames over ATM. |
| BN | Token Ring dedicated bridge number in a source route bridge environment. |
| Broadcast and Unknown Server (BUS) | BUS defines the set of functions that provide ELAN or LAN emulation transmission support while a switched virtual circuit connection is being established. It also supports LAN emulation broadcast services. |
| Emulated LAN | The emulation of the services of an Ethernet/IEEE 802.3 or Token Ring 802.5 LAN over an ATM network. |
| Interim Local Management Interface (ILMI) | Refers to ATM forum-defined interim specifications for network management functions between an end user and a public or private network, and between a public network and a private network. It is based on a limited subset of SNMP capabilities. |
| LAN emulation | Refers to the emulation of the connectionless nature of a LAN over connection-oriented ATM circuits. |
| LAN emulation client | Defines the set of functions implemented in an end system to interface with an ATM network in support of LAN emulation. |
| LAN emulation configuration server | Defines the set of functions that provide LECs with information regarding the location of the LAN emulation servers (LES). |
| LAN emulation server | Defines the set of functions that support ELAN registration and address resolution. |
| LAN emulation user network interface (LUNI) | Protocols allowing ATM-attached end systems and LAN/ATM conversion devices to control the virtual connections required for transmission and to emulate the connectionless nature of a LAN. |
| Permanent virtual circuit (PVC) | A virtual channel connection that has been established by manual or semi-automated methods. It is similar to a leased or dedicated real circuit. |
| RD | Token Ring route descriptor consisting of a Ring Number and a Bridge Number (RN,BN) in a source route bridge environment. |

| | |
|----------------------------------|---|
| RN | Token Ring dedicated ring number in a source route bridge environment. |
| STE packets | Spanning Tree Explorer packets used for Route Discovery or used to arrive at the destination end station through a single route in an SR LAN. |
| Switched virtual circuit (SVC) | A virtual channel connection that has been dynamically established in response to a signaling request message. |
| UNI Management Entity (UME) | The code residing in ATM devices at each end of a UNI interface that implements the management interface to the ATM network. |
| User-to-network interface (UNI) | ATM forum-developed specifications for the procedures and protocols between a user end station and the ATM network to effectively use ATM services and capabilities. |
| Virtual channel connection (VCC) | Virtual channels in two or more sequential physical circuits concatenated to create an end-to-end connection. A VCC is a specific instance of a switched virtual circuit (SVC) or a permanent virtual circuit (PVC). |
| Virtual channel identifier (VCI) | The 16-bit number in an ATM cell header identifying the specific virtual channel on which the cell is traversing on the current physical circuit. |
| Virtual path identifier (VPI) | The 8-bit number in an ATM UNI cell header identifying the specific virtual path on which the cell is traversing on the current physical circuit. |
| Virtual port | A logical attachment to a network. On a serial port, the attachment is made to a specific logical channel on the serial port. For example, in Frame Relay, the virtual port is configured on a specific DLCI over the Frame Relay port. For ATM, the virtual port is configured for the different types of ATM applications over the ATM port. There will be a Virtual Port for MPOA (MPATM), Classical IP over ATM (IPATM) or LAN Emulation Client over ATM (ETHATM or TRATM). |



CONFIGURING WIDE AREA NETWORKING USING THE ATM DXI

This chapter describes how to configure your bridge/router to establish serial line connectivity through the Asynchronous Transfer Mode data exchange interface (ATM DXI).



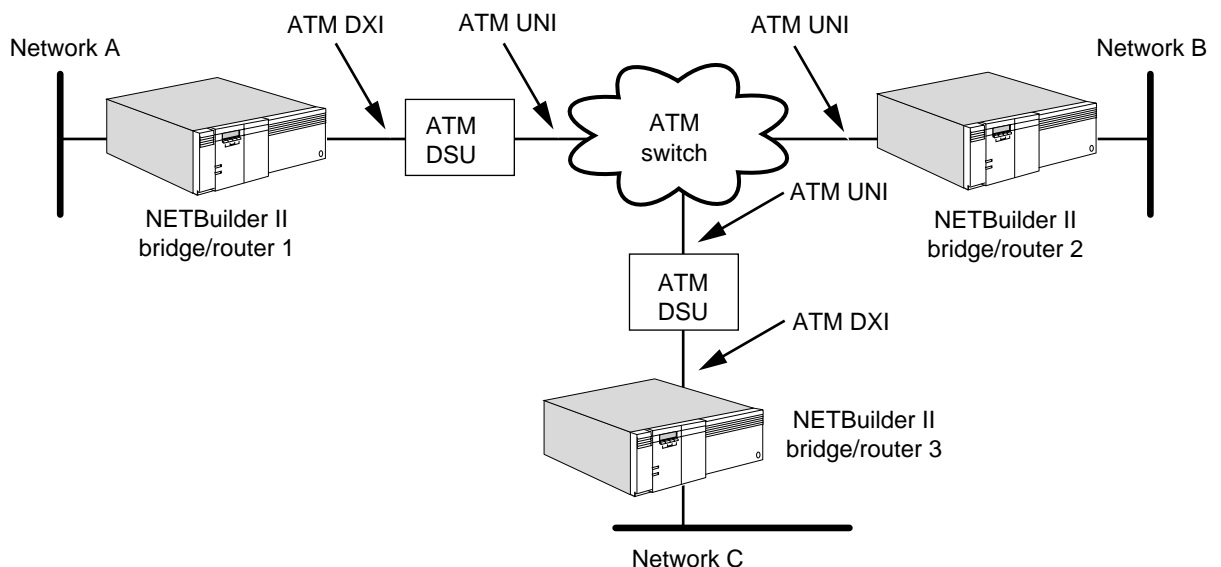
For conceptual information, see “How ATM DXI Works” later in this chapter.

The wide area bridge/router supports both bridging and routing of multiple protocols over ATM DXI. The ATM DXI software allows your bridge/router to transmit and receive data over a permanent virtual circuit (PVC) link with any other device on the ATM network, without requiring the installation of an ATM hardware module.

By using ATM DXI software in software version 8.0 or later, your bridge/router can access the ATM switch and network through an external ATM digital service unit (DSU). The ATM DSU segments and reassembles cells, provides the ATM adaptation layer (AAL3/4 or AAL5), and provides the user-to-network interface (UNI) needed to connect to the ATM switch (see Figure 433).

Your bridge/router acts as data terminal equipment (DTE), and the ATM DSU acts as data communications equipment (DCE). Bridge/routers from other vendors may attach to the ATM switch either through a DSU such as the NETBuilder II bridge/router, or directly through a UNI interface.

Figure 433 Typical ATM Connectivity



Configuring ATM DXI

Networking over ATM using the ATM DXI mode 1A is similar to networking over Frame Relay. You configure ATM DXI on the bridge/router as part of the 3Com FR Service, and all higher-level protocols use the Frame Relay configurations. To configure bridging and routing over ATM DXI, follow the procedures in the Configuring Wide Area Networking Using Frame Relay chapter, as if you were configuring a Frame Relay network. There are differences between ATM DXI and Frame Relay in addressing, higher-layer protocol encapsulation, and LMI Protocol features. These differences, and the corresponding changes in the configuration procedures, are explained in this section. You must consider these differences when you configure an ATM network.

Your bridge/router is also Frame-based UNI (FUNI) capable. FUNI is a variation of ATM DXI and is intended as a carrier service interface. A router currently running ATM DXI can successfully operate across a FUNI with no change. The ATM DSU is replaced with a conventional channel service unit/digital service unit (CSU/DSU), and the segmentation and reassembly function is moved into the carrier network.

ATM Address Mapping

In Frame Relay, PVCs are identified by 10-bit data link connection identifiers (DLCIs), usually represented as a decimal number between 0 and 1,023. You enter these DLCIs when you configure bridging and routing protocols, as described in the Configuring Wide Area Networking Using Frame Relay chapter.

In ATM, PVCs are identified by an 8-bit virtual path identifier (VPI) and a 16-bit virtual circuit identifier (VCI). The PVC is usually represented in VPI.VCI format, where VPI is a decimal number between 0 and 255 and VCI is a decimal number between 0 and 65,535.

You use the FR Service when you configure an ATM network on a NETBuilder II bridge/router, and you must enter addresses in Frame Relay format. The `AtmToFr` and `FrToAtm` utilities convert between the two address formats:

- The following syntax returns the decimal DLCI address corresponding to a decimal VPI.VCI address:

```
AtmToFr <vpi.vci> (0-255.0-65535)
```

- The following syntax returns the decimal VPI.VCI address corresponding to a decimal DLCI address:

```
FrToAtm <dldci> (0-1023)
```

Many different VPI.VCI addresses can map to a single DLCI. To avoid addressing errors, do not use multiple VPI.VCI addresses that map to the same DLCI.



Some vendors' DSUs require an ATM address that consists of a 0-bit VPI and a 10-bit VCI. In this case, the 10-bit VCI maps directly to a DLCI. You do not need the address conversion utilities with these addresses.

If your DSU vendor converts between VPI.VCI and DLCI addresses by bit mapping, use the address conversion utilities wherever the Frame Relay configuration procedures require a DLCI address. Otherwise, use the VCI portion of the VPI.VCI address directly as the DLCI address.

Encapsulation Type and AAL Support

In the procedure for “Setting Up the Frame Relay Service” in the Configuring Wide Area Networking Using Frame Relay chapter, add the following step to set the encapsulation type and provide ATM Application Layer (AAL) support:

Set the ATM mode for the physical port, or selectively on each virtual port, using:

```
SETDefault !<port> -FR AtmMode = {Enable | Disable, AAL34 | AAL5}
```

If the router at the other end of the virtual circuit supports LLC/SNAP encapsulation, enable ATM mode. This sets the encapsulation type to LLC/SNAP, the normal ATM mode. If the router does not support encapsulation, disable this mode. A NETBuilder II bridge/router running software prior to 8.0 or another vendor's router may not support encapsulation. This sets the encapsulation type to NLPID, the normal Frame Relay mode. The default is disabled. bridge/routers at both ends of a virtual circuit must use the same encapsulation type for successful operation.

Use the AAL34 parameter when connecting to an ATM DSU that supports only ATM Adaptation Layer AAL3/4. Use AAL5 when connecting to a DSU that supports AAL5. The default is AAL5. Bridge/routers at both ends of a virtual circuit must use the same adaptation layer.

LMI Protocol

ATM DXI supports an LMI Protocol that is very different from the LMI Protocol used with Frame Relay. NETBuilder II bridge/routers do not support the ATM DXI LMI Protocol. This difference causes the following changes in the configuration procedure.

Setting Up the ATM Service

In the procedure for “Setting Up the Frame Relay Service” in the Configuring Wide Area Networking Using Frame Relay chapter, you must disable the Frame Relay LMI Protocol in step 2, using:

```
SETDefault !<port> -FR CONTrol = NoLMI
```

Step 3 of this procedure then becomes unnecessary.

Configuring Transparent Bridging

Because ATM does not support Frame Relay LMI, you must configure transparent bridging by manually adding DLCI neighbors to the static DLCI neighbor table. This procedure is explained in step 1 under “Configuring Wide Area Networking Using Frame Relay” in the Configuring Wide Area Networking Using Frame Relay chapter.

Remember to convert the neighbors' VPI.VCI addresses to DLCI format, if necessary, using the AtmToFr utility.

Configuring IPX over an ATM Network

Because ATM does not support LMI, you must manually enter mapping information between the ATM addresses and host addresses for each bridge/router directly connected to the ATM network. This procedure is explained in step 2 of “Configuring IPX” in the Configuring Wide Area Networking Using Frame Relay chapter.

Configuring XNS over an ATM Network

Because ATM does not support LMI, you must manually enter mapping information between the ATM addresses and host addresses for each

bridge/router directly connected to the ATM network. This procedure is explained in step 2 of "Configuring XNS" in the Configuring Wide Area Networking Using Frame Relay chapter.

How ATM DXI Works

This section explains the differences between ATM and Frame Relay in address mapping and encapsulation type.

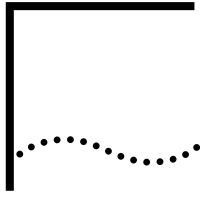
Address Mapping

The PVC addresses that the user obtains from the ATM switch usually are in VPI.VCI format. These addresses must be converted into DLCI format in order to configure higher-level protocols according to the procedures in the Configuring Wide Area Networking Using Frame Relay chapter.

Enterprise OS software provides the `AtmToFr` and `FrToAtm` utilities to convert between the two address formats. For further information about VPI.VCI and DLCI formats and the conversion utilities, see "ATM Address Mapping" earlier in this chapter.

Encapsulation Type

In Frame Relay, higher-layer protocols are encapsulated using the one-byte Network Layer Protocol Identifier (NLPID) specified by RFC 1490. In ATM they are normally encapsulated using the logical link control/Subnetwork Access Protocol (LLC/SNAP) method defined in RFC 1483. If you need connectivity between a NETBuilder II bridge/router running ATM and another router that supports Frame Relay but not ATM (such as a NETBuilder II bridge/router running software prior to 8.0, or another vendor's router), you can set the encapsulation type to NLPID by disabling the `-FR AtmMode` parameter.



CONFIGURING FDDI

This chapter describes the following information on configuring the Fiber Distributed Data Interface (FDDI):

- Port configuration for FDDI usage
- FDDI maintenance and troubleshooting information

Configuring Ports for FDDI

When an FDDI board is installed in your system, the software automatically sets the port ownership for the corresponding port to FDDI. FDDI port configuration is transparent and no additional user-configuration activity is required.

Troubleshooting the Configuration

If you have problems making FDDI connections to other networks after setting up your router, review the following troubleshooting procedure. This procedure can help you diagnose various network and internal hardware problems. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier for assistance. For more information on the FDDI commands and parameters discussed in this chapter, see the Commands chapter and the FDDI Service Parameters chapter in *Reference for Enterprise OS Software*.

Diagnosing Internal Hardware Problems

If both PHY LEDs on the media access control (MAC) board do not light green after you initialize the system and connect to an operational ring, you need to perform a self-test.

To perform the self-test, follow these steps:

- 1 Remove the two connections from your station to the network, then loop PHY port A to PHY port B using a length of standard media interface connector/media interface connector (MIC/MIC) fiber-optic cable.

This connection puts your station into loopback mode. If both PHY LEDs are green, the FDDI interface on your station is operating normally and the problem exists elsewhere, either with a neighbor station or with the line itself. Steps 3 through 5 describe how to perform line-state testing.

If either of the two PHY LEDs are red while in loopback mode, your PHY board is defective and should be replaced.

- 2 Remove the loopback connection made in step 1 and reconnect your station to the network.
- 3 To perform line-state testing for each port, first set ports A and B to maintenance state using the PControl parameters:

```
SET !<path> -FDDI PControlA = Maint [sets port A]  
SET !<path> -FDDI PControlB = Maint [sets port B]
```

- 4 Display the current line states using the Maintenance Line State parameters:

```
SHow !<path> -FDDI MaintLineStateA
```

```
SHow !<path> -FDDI MaintLineStyleB
```

These commands display the line state of the transmitter first, then the line state of the receiver.

- 5 Use the Idle and Halt values of the MaintLineStyle parameters to conduct additional tests of the lines:

```
SET !<path> -FDDI MaintLineStyleA = Idle
SET !<path> -FDDI MaintLineStyleB = Idle
SET !<path> -FDDI MaintLineStyleA = Halt
SET !<path> -FDDI MaintLineStyleB = Halt
```

These commands cause port A or B to transmit Idle or Halt symbols continuously.

If the line states being transmitted match the received line states at the other end of the fiber-optic cable for both directions, the fiber-optic transceivers at each end are in normal working order and are compatible. This is only a static test and does not diagnose an intermittent component failure.

Diagnosing Network Problems

Use the following FDDI parameters to help diagnose network problems:

- Use the PortNeighbor parameter to determine whether an undesirable connection is in operation, such as port A attempting to communicate with a port A neighbor.

For normal dual-attachment operation, the neighbor for port A should be port B, and the neighbor for port B should be port A. For example, to display the port A and port B neighbor types for path 2, enter:

```
SHow !2 -FDDI PortNeighbor
```

A message similar to the following appears:

```
Port A: PortNeighbor = B
Port B: PortNeighbor = A
```

- Use the SMTAddress, UpNeighbor, and DownNeighbor parameters to determine the MAC address (12 hex characters) of your 3Com bridge/router FDDI station and of your neighbors on the ring.

For example, to display the MAC addresses of the upstream and downstream neighbors for path 2, enter:

```
SHow !2 -FDDI UpNeighbor DownNeighbor
```

- Use the DupAddress parameter to determine if your station has detected a duplicate MAC address on the FDDI ring.

The DupAddress parameter displays the setting of the duplicate address flag. The duplicate address flag is set to the Detected state when a frame is detected with a MAC address that is a duplicate of the MAC address of the receiving station. For example, to display the duplicate address flag setting for path 2, enter:

```
SHow !2 -FDDI DupAddress
```

- Use the StationCONFig parameter to determine if the station is in a wrap or through state.

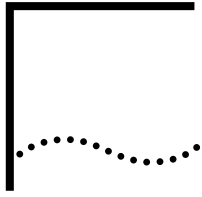
A wrap state occurs when one of the fiber-optic links is operational and one has failed, or when a port is connected to a concentrator. For example, to display the current configuration state of your station, enter:

```
SHow !1 -FDDI StationCONFig
```

- Use the RemDisconnect parameter to determine if a disconnect was requested by a remote management station, when a station disconnects automatically (all LEDs red).

The RemDisconnect parameter displays the current value of the Remote Disconnect Flag. When set (value = yes), this flag indicates that the station has been remotely disconnected. For example, to display the current value of the Remote Disconnect Flag for path 1, enter:

```
SHow !1 -FDDI RemDisconnect
```

CONFIGURING MNEMONIC FILTERING

This chapter describes the procedures for configuring filters and also lists all the built-in masks for the bridge and Internetwork Packet Exchange (IPX) router. Filtering is an operation that determines whether specified packets are forwarded or discarded by your 3Com bridge or IPX router. The Filter Service also controls these and other capabilities through the Filter POLicy parameter action options: Count, Discard, DoDiscard, Forward, PROTOcolRsrv <tag>, Sequence, Prioritization, and Trace. These action options are described in “Action” later in this chapter.

You need to configure prioritization separately. For complete information on the prioritization allocation, see the Prioritizing Multiprotocol Data chapter.

By using filtering in a bridged or IPX routed environment, you can:

- Achieve security and bandwidth protection by isolating specific segments of the network.
- Monitor network traffic by gathering statistics.
- Adjust the performance of your network to fit the traffic flow.
- Sequence packets so that they are received in the order they were sent.
- Reserve bandwidth for particular protocols, so that large-bandwidth user applications, such as file transfer and mail, share link capacity with lower bandwidth users such as interactive sessions and transaction-oriented applications.

Enterprise OS software includes the use of mnemonics and built-in masks for specific protocols in the configuration of filters. Through the use of built-in mnemonics, you can also create user-defined masks to meet more specialized needs.

For more information on the parameters used in creating filters and masks, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*. For conceptual information, see “How Filtering Works” later in this chapter.

Configuring Filters

When you configure filters, you perform the selection, qualification, and action steps using the ADD and DELeTe commands. The MASK parameter specifies the selection criteria and the POLicy parameter specifies the context by qualifying the selection and associating the action.

You can use the same selection criteria (masks) in different contexts (policies). You can also combine different selection criteria while qualifying them and specifying the action. The procedures in this section use the minimum number of steps required to configure basic built-in and user-defined filters for the bridge.

Using Built-in Masks To configure filters for the bridge or IPX router using built-in masks, follow these steps:

- 1 Determine whether or not a built-in mask can be used as follows:
 - a Identify the type of packet to be filtered.
 - b After identifying the packet type, see Table 98 for the BRIDGE Service, or Table 99 for the IPX Service, or Table 100 for IBM Trace built-in masks.

These tables identify all types of packets for which built-in masks can be used. If a built-in mask can be used, proceed to step 2. If a built-in mask cannot be used, follow the steps in “Using User-defined Masks” next.

- 2 Define the policy by using the ADD POLICY command.

Add a policy whether or not the mask is built-in.

For example, suppose you want to discard all Internet Protocol (IP) multicast packets at port 2. To define the policy, enter:

```
ADD -Filter POLICY NoIPMC Discard IP MC AT !2
```

The following message appears on the screen:

```
Policy NoIPMC is added
```

Continue using the ADD MASK and ADD POLICY commands for all types of packets to be filtered.

- 3 Specify the action for packets that do not match any policy by setting the DefaultAction parameter:

```
SETDefault -Filter DefaultAction = [Forward | Discard]
```

When DefaultAction is set to Discard, all packets not matching a policy are discarded. All packets matching the policy are handled according to the policy.

- 4 Enable filtering by entering:

```
SETDefault -Filter CONTROL = Enabled
```

Using User-defined Masks To configure a filter using user-defined masks, follow these steps:

- 1 Determine whether or not a built-in mask can be used as follows:
 - a Identify the type of packet to be filtered.
 - b After identifying the packet type, see Table 98 for the BRIDGE Service or Table 99 for the IPX Service.

These tables identify all types of packets for which built-in masks can be used. If a built-in mask can be applied, follow the steps in “Using Built-in Masks” earlier in this chapter. If a built-in mask cannot be applied, proceed to step 2.
- 2 If a built-in mask cannot be used, and built-in mnemonics is supported, define your own mask by using the ADD MASK command.

Table 101 and Table 102 list the built-in mnemonics that can be used to construct user-defined masks for the BRIDGE and IPX Services.

Suppose you want to define a pattern for a mask that is not built-in (that is, not represented in Table 98 or Table 99). For example, you may want to discard all packets that are longer than 512 bytes. Because you cannot represent this pattern as a built-in mask, you must enter the following command and the built-in mnemonics (dl.length) to define the mask:

```
ADD -Filter MASK longpkts dl.length>%0200
```

The following message appears on the screen:

```
Mask LONGPKTS is added
```



The expected value must be an even number of digits.

- 3 Define the policy by using the ADD POLicy command.

Add a policy whether or not the mask is built-in. For example, suppose you still want to discard all packets that are longer than 512 bytes at port 2, as in step 2. You have defined the mask. To define the policy, enter:

```
ADD -Filter POLicy toolong Discard longpkts AT !2
```

The following message appears on the screen:

```
Policy TOOLONG is added
```

- 4 Specify the action of the packet that does not match any policy by setting the DefaultAction parameter using:

```
SETDefault -Filter DefaultAction = [Forward | Discard]
```

When DefaultAction is set to Discard, all packets that do not match a policy are discarded. All packets that match the policy are handled as designated in the policy.

- 5 Enable filtering by entering:

```
SETDefault -Filter CONTrol = Enabled
```

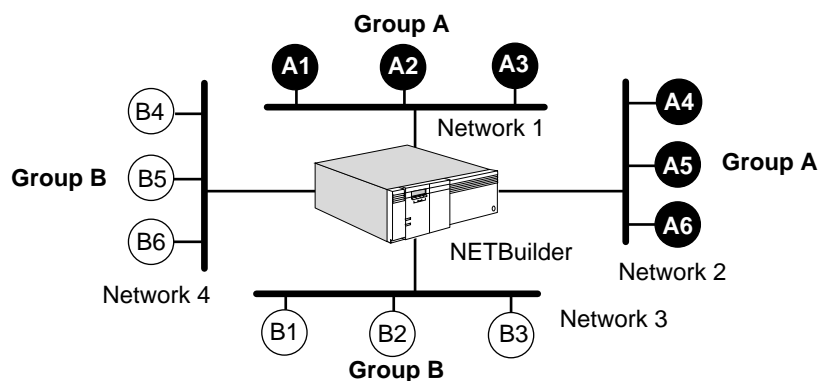
Grouping Related Stations

To configure a filter for a group of logically related stations, use the StationGroup parameter. When using the StationGroup parameter, you need to complete the following tasks:

- Assign a set of station addresses for easy reference.
- Give the group a name.
- Create a mask by referencing the station group name.

Example Figure 434 is an example of specifying a policy based on station groups.

Figure 434 Network Showing Station Groups



In this figure, stations belong to group A or to group B. Group A has stations on network 1 and network 2. Group B has stations on network 3 and network 4. After grouping the stations, you can create a policy that would, for example, prohibit a certain type of traffic between group A and group B. Assuming that the media access control (MAC) address for station A1 is %0800020000a1 and the MAC address for station A2 is %0800020000a2, follow these steps to configure a filter between group A and group B:

- 1 Define a station group and add the MAC addresses of the stations belonging to the defined group.

For example, create group A and group B, and add appropriate addresses to them by entering:

```
ADD -Filter StationGroup group_a %0800020000a1
ADD -Filter StationGroup group_a %0800020000a2
ADD -Filter StationGroup group_a %0800020000a3
ADD -Filter StationGroup group_a %0800020000a4
ADD -Filter StationGroup group_a %0800020000a5
ADD -Filter StationGroup group_a %0800020000a6
ADD -Filter StationGroup group_b %0800020000b1
ADD -Filter StationGroup group_b %0800020000b2
ADD -Filter StationGroup group_b %0800020000b3
ADD -Filter StationGroup group_b %0800020000b4
ADD -Filter StationGroup group_b %0800020000b5
ADD -Filter StationGroup group_b %0800020000b6
```

- 2 Define masks using the station groups.

For example, to create masks, enter:

```
ADD -Filter MASK from_group_a DataLink.SrcAddr = group_a
ADD -Filter MASK from_group_b DataLink.SrcAddr = group_b
ADD -Filter MASK to_group_a DataLink.DstAddr = group_a
ADD -Filter MASK to_group_b DataLink.DstAddr = group_b
```

- 3 Define policies using the previously defined masks.

For example, to create policies, enter:

```
ADD -FI POLICY block_from_a Discard from_group_a, to_group_b, IP
ADD -FI POLICY block_from_b Discard from_group_b, to_group_a, IP
```

For more information on the StationGroup parameter, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Parameter Overview

Table 97 lists and briefly describes the Filter Service parameters. For detailed descriptions of these parameters, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Table 97 Filter Service Parameters

| Parameter | Description |
|-------------------|---|
| CONFigurati
on | Displays the overall configuration of the Filter Service. |
| CONTRol | Disables and enables the Filter Service. Must be enabled for any filter-related actions to be performed. |
| DefaultActio
n | Specifies the action applied to a packet if it does not match any of the policies configured. (If default is altered to Discard, and there are no forwarding policies defined, no packets are forwarded by the system.) |
| (continued) | |
| DIAGnostics | Shows the current decision tree that the system is using. Shows which MASKs are associated with which POLicies. |
| MASK | Defines the criteria used to select a packet for special handling. |
| MNEmonics | Displays all possible options for a location that can be used to construct a user-defined mask. |
| POLicy | Defines the system context within which the specified masks are applied and the action to be taken. Uses the MASKs that are defined, and applies specific operations to packets that match the MASK conditions of the POLicy. |
| SElection | Lists all services for which the filter function can be invoked (BRIDGE, IPX, DLSW, LLC2 or SDLC). |
| StationGroup | Groups a set of station addresses for easy reference. |

How Filtering Works

This section explains the filtering process.

A filter contains the following two components:

- A mask, which defines the qualifications a packet must meet
- A policy, which defines which masks are to be applied and what action is to be taken for the packets that meet the criteria of the mask

For packets using filters based on either user-defined masks or built-in mnemonic masks, the following Filter Service POLicy parameter action options are available: Count, Discard, DodDiscard, Forward, PROTOcolRsrv <tag>, Sequence, Prioritization, and Trace.

When you use filters with user-defined masks, you need to determine location offsets and values to create the mask. Using built-in masks allows you to specify packet selection criteria without determining specific offsets, encapsulation, and frame formats. These built-in masks simplify filtering operations for the bridge and make filtering configurations transferable across interfaces of different types. Most

built-in masks are defined for specific protocols. These masks are listed later in this chapter.

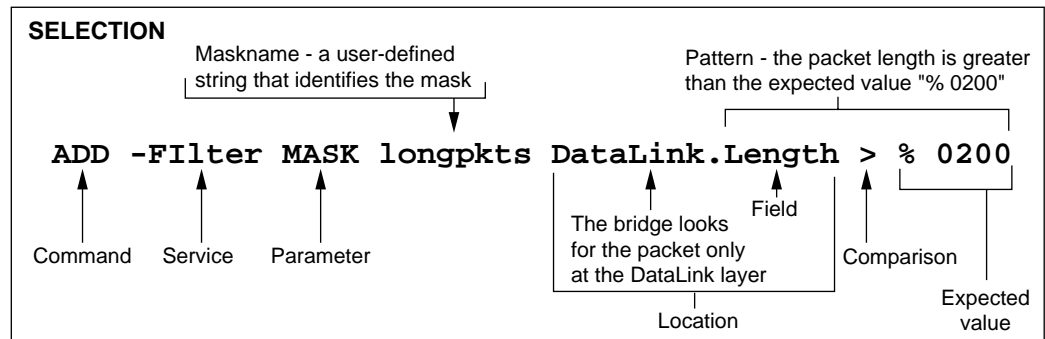
To support user-defined masks, Enterprise OS software has several built-in mnemonics that can be used to specify location and pattern. The locations and patterns are listed later in this chapter.

The filtering operation involves the steps of selection, qualification, and action.

Selection Selection identifies the packets on which filtering is performed. You can select packets for special action by specifying a particular pattern of data at a particular location. You can also specify other, more complicated, selection criteria. Use the MASK parameter to select the packet.

Figure 435 is an example of the use of MASK parameter in the selection process. The location is typically specified as a string of hexadecimal numbers. In Figure 435, the use of built-in BRidge mnemonics lets you specify the location at the DataLink layer. The offset for the same field within a packet can vary, depending on the encapsulation or frame format. For more information on the MASK parameter, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Figure 435 Filter Selection Process

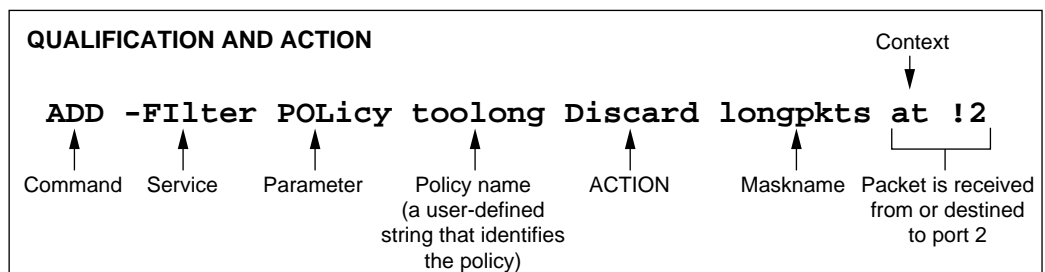


Qualification Qualification specifies the context of the filtering operation, that is, the direction of travel and the ports affected.

After selecting a packet for special action using the MASK parameter, you may specify additional qualifications before the action is taken. For example, using Enterprise OS software, it is possible to select only those broadcast packets that arrive on a specified port, instead of all broadcast packets. Use the POLICY parameter to specify qualifications for the packet.

Action After the packet is selected and qualified, a specified action occurs. Use the POLICY parameter in the Filter Service to specify the desired action. The action options supported in the Enterprise OS software are Count, Discard, DodDiscard, Forward, PROTOcolRsrv <tag>, Sequence, Prioritization, and Trace. Figure 436 illustrates the qualification and action processes using an example of the POLICY parameter.

Figure 436 Filter Qualification and Action Processes



Count

When you use the Count option, you count packets that meet specified criteria. For example, you may want to count all IP packets forwarded by the bridge before deciding how the bridge should handle them. To perform this operation, enter:

```
ADD -Filter POLICY IP_count Count ip
```

Discard

When you use the Discard option, you can discard packets that match specific criteria.

DodDiscard

When you use the DodDiscard option for a dial-on-demand (DOD) port, if the dial-up path is down, you can ensure that the packet is discarded and does not

cause the dial-up path to be raised. If the path is up, the packet is forwarded, but is not considered as user traffic that keeps a dial-up path up.

Forward

Filters can prevent packets meeting certain criteria from being forwarded across the system or forward only those packets meeting specified criteria while blocking all others. When you use the Forward option, you forward packets that match specific criteria. For more information on forwarding, see the POLicy parameter in the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

PROTOCOLRsrv <tag>

Protocol reservation assigns a specified percentage of bandwidth to designated packets passing through a specified port and meeting specified conditions. The specified conditions can be protocol type, packet length, packets destined for specified address, and so on.

Protocol reservation is set up with different procedures depending on the packet types being configured for protocol reservation. The mnemonic filtering procedure applies to all bridged packets and all IPX-routed packets. The IP filtering procedure applies only to IP-routed packets. IP-routed packets are also filtered using the IP firewall feature. See the Building Internet Firewalls chapter for detailed information about the IP firewall feature.

For a detailed description of the protocol reservation procedures for all the packet types, see the Configuring Protocol Reservation chapter.

As part of the mnemonic filtering procedure, you enter the PROTOCOLRsrv <tag> action option to apply protocol reservation to designated packets. The tag name identifies those packets that receive a specified percentage of bandwidth when passing through the specified WAN port and when meeting the mask conditions set up with the Filter Service POLicy parameter. Tag the designated packets with the identifying name by entering a name as the <tag> value when you enter the PROTOCOLRsrv <tag> action option. The tag name can be any alphanumeric string no longer than 15 characters.

For bridge filtering examples using the PROTOCOLRsrv <tag> action option and the -PORT PROTOCOLRsrv parameter, see Example 26, Example 27, and Example 28 in "Bridge Filtering Examples" later in this chapter.

For an IPX filtering example using the PROTOCOLRsrv <tag> action option and the -PORT PROTOCOLRsrv parameter, see Example 9 in "IPX Filtering Examples" later in this chapter.

Sequence

You can sequence packets to ensure that they arrive at their destination in the order they were sent. To ensure that packets arrive in sequence, use the Sequence option. When the load-balancing algorithm is operating, packets can arrive out of sequence.

When operating with two or more parallel lines (including bandwidth-on-demand dial-up lines), local area transport (LAT), NETBEUI, and Logical Link Control type 2 (LLC2) should be packet-sequenced using the sequence policies. If all of the traffic

on the port is sequenced, bandwidth-on-demand is not used for that data. Sequenced traffic is only sent on the primary path.

For example, if you want to sequence and send LAT packets to port 4 in the order they are received, enter:

```
ADD -Filter POLIcy LATorder Sequence LAT TO !4
```

For more information on sequencing and the POLIcy parameter, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Prioritization (Priority Queuing)

The Prioritization option allows you to prioritize different packet types transmitted over wide area networks. You can assign priorities to packets according to their protocol type. Prioritization is a filtering component and needs to be configured separately. For complete information on data prioritization, see the Prioritizing Multiprotocol Data chapter.

Trace

You can trace packets from IBM-related protocols such as APPN, DLSw, LLC2, and SDLC. You can use these traces to determine the status of connections and to isolate problems. The Trace option cannot be used for any other type of packet.

For a more detailed explanation of the -Filter MASK and -Filter POLIcy parameters, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Built-in Bridge Masks

Enterprise OS software supports several built-in predefined selection criteria, or masks. All bridge masks are associated with DataLink level as the protocol, and all IPX built-in masks are associated with IPX as the protocol. Table 98 lists the built-in DataLink masks. To display this table, enter:

```
SHow -Filter MASK BuiltIn
```

Table 98 Built-in Bridge Masks

| Built-in Mask | Equivalent | Packet Type |
|---------------|------------------------------------|--------------|
| BC | DataLink.DestinationAddr=BroadCast | Bcast |
| MC | DataLink.DestinationAddr=MultiCast | Mcast |
| ATALK | DataLink.Protocol=AppleTalk | AT |
| AARP | DataLink.Protocol=AARP | AppleTalkARP |
| ARP | DataLink.Protocol=ARP | ARP |
| CLNP | DataLink.Protocol=CLNP | OSI-related |
| DECNET | DataLink.Protocol=DECnet | DECnet |
| DLTEST | DataLink.Protocol=DLTest | DLTest |
| IP | DataLink.Protocol=IP | IP |
| IPX | DataLink.Protocol=IPX | Novell IPX |
| LAT | DataLink.Protocol=LAT | LAT |
| (continued) | | |
| NMIP | DataLink.Protocol=NetMapIP | NetMapIP |
| NMXNS | DataLink.Protocol=NetMapXNS | NetMapXNS |

Table 98 Built-in Bridge Masks (continued)

| | | |
|-------|--|---------------------------|
| STP | DataLink.Protocol=STP | Spanning Tree |
| VIP | DataLink.Protocol=VIP | VINES |
| XNS | DataLink.Protocol=XNS | XNS |
| SR | DataLink.RoutingType=SpecificRoute | Specifically Routed Frame |
| SRE | DataLink.RoutingType=SingleRouteExplorer | Spanning Tree Explorer |
| ARE | DataLink.RoutingType=AllRouteExplorer | All Route Explorer |
| ALLRT | DataLink.RoutingType=ALL | Any source-routed frame |

Built-in IPX Masks

Table 99 lists the built-in IPX masks. These predefined masks identify different types of IPX packets. To display this table, enter:

```
SHow -Filter MASK Builtin
```

Table 99 Built-in IPX Masks

| Built-in Mask | Use |
|---------------|---|
| IPXRIP | Matches a RIP packet. |
| SAP | Matches a SAP packet. |
| FSP | Matches a Netware File Service NCP packet. |
| WANBC | Matches a broadcast packet of IPX packet type 20. |
| TRACERT | Matches a 3Com-proprietary Trace packet (soc = 0x874e). |
| IPXPING | Matches an IPX Ping packet (soc = 0x9086). |
| IPXDIAG | Matches an IPX Diagnostic packet (soc = 0x456). |
| NWSEC | Matches a Netware Security packet (soc = 0x457). |

Built-in IBM Trace Masks

Table 100 lists the built-in IBM Trace masks. For more information about using the IBM Trace facility, see the IBM Trace Facility appendix.

Table 100 Built-in IBM Trace Masks

| Built-in Mask | Equivalent | Packet Type |
|---------------|------------------------|--------------------------|
| LLC2 | Datalink.Protocol=LLC2 | LLC2 |
| SDLC | Datalink.Protocol=SDLC | SDLC |
| DLSW | Datalink.Protocol=DLSW | DLSW |
| DLCTL | DLSW.1 = 72 | DLSW Control Message |
| DLSWI | DLSW.1 = 16 | DLSW Information Message |

User-defined Bridge Masks

When you use the ADD MASK command, you must specify a location. The location is usually expressed as a hexadecimal value representing the offset from the beginning of a packet at which a specified pattern of data is compared to the

contents of a packet. The packet is selected if it matches the pattern of data at the specified location.

You also can specify a location in the mnemonic form: <protocol>.<field>. This format allows encapsulation-independent relative offsets to be used. You do not need to determine frame formats or specific offsets. All bridge mnemonics are associated with DataLink as <protocol>. Different mnemonic values are allowed for the <field> and <match> locations. To support IPX filtering, a set of IPX-specific mnemonics is provided. All IPX mnemonics are associated with IPX as <protocol>.

Table 101 shows valid locations that match the DataLink protocol. Use these fields to specify an address, instead of specifying the offset of a particular field.

To display a list of valid locations supported by the bridge, enter the `SHoW -Filter MNEMonics` command. ALL is a valid match mnemonic for certain field categories. When ALL is specified, any value in the location is considered to match the criteria. Field mnemonics indicate encapsulation-independent relative offset. The software recognizes the encapsulation and locates the <field> at the correct offset.

Table 101 User-defined Bridge Masks and DataLink Locations

| Field | Description | Matching Value |
|--------------------------------|--|--|
| DstAddr | Destination Address at DataLink layer | <MAC address>
ALL
<StationGroup> |
| SrcAddr | Source Address at DataLink layer | <MAC address>
<StationGroup> |
| Address | Either Destination or Source Address at DataLink layer | <MAC address>
<StationGroup> |
| Protocol | Packet protocol type | <numerical value> |
| LENGth | Frame size, including padding | <numerical value> |
| DSAP | Destination service access point | <numerical value> |
| SSAP | Source service access point | <numerical value> |
| LSAP | Link service access point, destination or source SAP | <numerical value> |
| OUI | Organizationally unique ID | %<hexadecimal number> |
| LanID | LAN identifier in a source-routed frame | <numerical value> |
| DATA+[%]<offset>[:[%]<length>] | Offset from start of DataLink data | %<hexadecimal number>
<numerical value> |
| [%]<offset>[:[%]<length>] | Offset from start of DataLink header | %<hexadecimal number>
<numerical value> |



The SR bit in the SourceAddress field of a source-routed frame is ignored during comparison.

User-defined IPX Masks

Table 102 lists user-defined IPX masks and valid locations. You can use these fields to specify an address, instead of specifying the offset of a particular field. ALL is a valid match mnemonic for certain field categories. When ALL is specified, any value in the location is considered to match the criteria. The % sign is used to enter hexadecimal values.

To display a list of valid locations supported by the Internetwork Packet Exchange (IPX) router, enter:

```
SHow -Filter MNEmonics
```

Table 102 IPX Built-in Mnemonics for User-defined Masks

| Field | Description | Matching Value |
|-------------|---|---|
| DsrNETwork | IPX destination network | <network number> |
| SrcNETwork | IPX source network | <network number> |
| NETwork | Either IPX destination or source network | <network number> |
| DstNodeAddr | IPX destination node address | %<host address> |
| SrcNodeAddr | IPX source node address | %<host address> |
| NodeAddr | Either IPX destination or source node address | %<host address> |
| DstSockeT | IPX destination socket | FileServicePacket
ServiceAdvertisingPacket
RoutingInformationPacket
IpxPingPacket
IpxDiagPacket
IpxTraceRoute
NWSecurityPacket
%<hexadecimal value>
<numerical> |
| SrcSockeT | IPX source socket | FileServicePacket
ServiceAdvertisingPacket
RoutingInformationPacket
IpxPingPacket
IpxDiagPacket
IpxTraceRoute
NWSecurityPacket
%<hexadecimal value>
<numerical> |

Table 102 IPX Built-in Mnemonics for User-defined Masks (continued)

| Field | Description | Matching Value |
|--|---|---|
| Socket | Either IPX destination or source socket | FileServicePacket
ServiceAdvertisingPacket
RoutingInformationPacket
IpxPingPacket
IpxDiagPacket
IpxTraceRoute
NWSecurityPacket
%<hexadecimal value>
<numerical value> |
| PacketLength | IPX packet length | %<hexadecimal value>
<numerical> |
| PacketType | IPX packet type | %<hexadecimal value>
<numerical> |
| TransportCtl | IPX transport control | %<hexadecimal value>
<numerical> |
| DATA+[%]<offset set>
[:[%]<length>] | Starting <offset> bytes after the end of the IPX header and <length> bytes long | %<hex num string>
<" ascii string" > |

Bridge Filtering Examples

This section contains examples of bridge filtering features. Examples of configuring the prioritization component of filtering are provided in the Prioritizing Multiprotocol Data chapter.

Example 1 To enable filtering and to stop checking policies after a policy that matches the packet is found, use:

```
SETDefault -Filter CONTROL = (Enabled, MatchOne)
```

Example 2 **Displaying all masks** To display all masks, enter:

```
SHow -Filter MASK
```

Displaying built-in masks To display all built-in masks, enter:

```
SHow -Filter MASK BuiltIn
```

Displaying a specific mask To display a specific mask, use:

```
SHow -Filter <maskname>
```

Example 3 **Displaying all policies** To display all policies, enter:

```
SHow -Filter POLicy
```

Displaying a specific policy To display a specific policy, use:

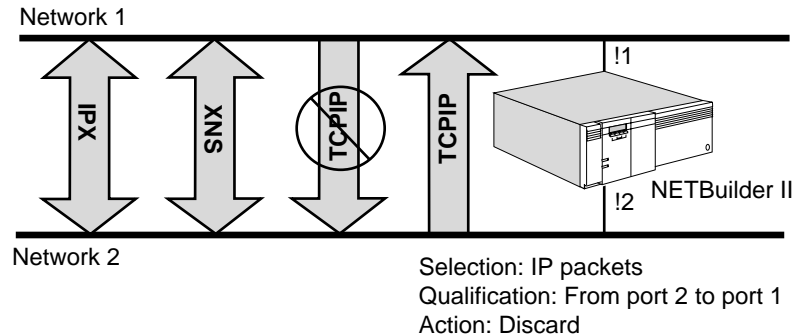
```
SHow -Filter <policyname>
```

Example 4 To discard all source-routed IP packets, enter:

```
ADD -Filter POLicy dissr_ip Discard ip, allrt
```

Example 5 This example describes how to discard all IP packets from port 1 to port 2 using two options: the command syntax and the menu (see Figure 437). IP packets are selected for special action. The selection is further qualified by specifying from port 1 to port 2. The action is designated as discard. Because built-in masks are defined for IP packets, it is not necessary to use the ADD MASK command.

Figure 437 Discarding IP Packets



Command Syntax Option Define the policy by entering:

```
ADD -Filter POLicy noip Discard ip FROM !1 TO !2
```

Menu Option You can use the Filter Service menu to discard all IP packets from port 1 to port 2. After entering the Filter Service, select the POLicy option of the Level 2 menu. The following screen appears:

```

=====Show -Filter POLicy=====
No policy defined
=====Filter POLicy parameter menu (Level 3)=====
      1 - Add
      2 - Delete
      3 - Flush
Select (1-3) ... <CR> to Exit =====> 1
    
```

Select 1. When the following screen appears, enter the policy "noip Discard IP FROM!1 to !2."

```

=====Show -Filter POLicy=====
No policy defined
=====Filter POLicy parameter menu (Level 3)=====
      1 - Add
      2 - Delete
      3 - Flush
Select (1-3) ... <CR> to Exit =====> 1
Add POLicy <polycname> <action> <masks> [<context>]
Add POLicy noip discard ip from !1 to !2
    
```

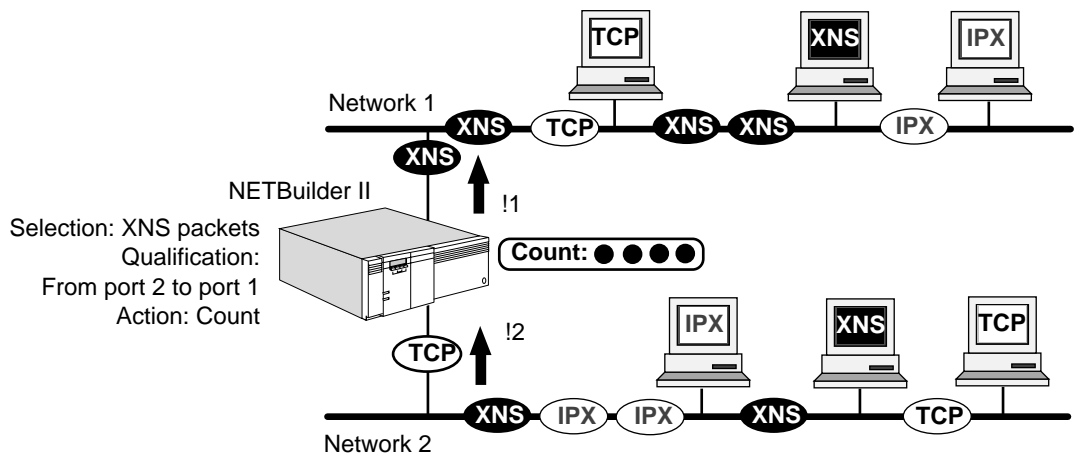
After the policy is added, the message "Policy noip is added" appears on the screen. The following screen now appears:

```

=====Show -Filter POLicy=====
1 policy defined
      id      name      action      masks
=====
      p0      NOIP      Discard IP  FROM !1 TO !2 (0, 0)
=====Filter POLicy parameter menu (Level 3)=====
      1 - Add
      2 - Delete
      3 - Flush
Select (1-3) ... <CR> to Exit =====> 1
[4]NETBuilder #
    
```

Example 6 Figure 438 shows how to count all XNS packets from port 2 to port 1.

Figure 438 Counting XNS Packets



Xerox Network Systems (XNS) packets are selected for special action. The selection is further qualified by specifying from port 2 to port 1. The action is designated as Count. Because built-in masks are defined for XNS packets (see Table 98), you only need to use the ADD POLicy command to define the policy.

The policy is added after you enter:

```
ADD -Filter POLicy xnspace Count xns FROM !2 TO !1
```

Example 7 **Discarding Packets on All Ports** To define a filter to discard DECnet packets on all ports, you need not define a mask, because a predefined mask for DECnet exists. This example could be used for any built-in mask by replacing the mask DECnet with the built-in mask that fits your need.

To define a DECnet filter for all ports, enter:

```
ADD -Filter POLicy discard_dec Discard decnet
```

Discarding Packets on a Specific Port To filter out DECnet packets at ports 2 and 3, enter:

```
ADD -Filter POLicy discdec Discard decnet AT !2, !3
```

Example 8 To check all policies, enter:

```
SHow -Filter POLicy
```

Example 9 To add a mask that selects packets destined to %080002123456, enter:

```
ADD -Filter MASK to_atlas dl.dstaddr = %080002123456
```

Example 10 To add a mask that selects packets with LLC encapsulation, enter (the value of either DSAP or SSAP is %aa.):

```
ADD -Filter MASK snap dl.dsap = %aa
```

Example 11 To add a mask that selects packets with a value greater than %45 at the first byte of data, enter:

```
ADD -Filter MASK some_data dl.data+%0>%45
```

After the mask is added, the message "Policy some_data is added" appears on the screen.

Example 12 To bridge IP traffic among ports 1, 3, 5, and 6, you can use either command A or B. Command A is preferred, because the built-in mask is encapsulation-independent. Command B forwards IP packets with Ethernet II encapsulation. However, IP packets from token ring or FDDI are handled incorrectly.

Command A:

```
ADD -Filter POLicy ipgroup Forward ip AMONG !1, !3, !5, !6
```

Command B:

```
ADD -Filter MASK ethernet_ip %c = %0800
```

```
ADD -Filter POLicy ipgroup Forward ethernet_ip AMONG !1, !3, !5, !6
```

Example 13 To isolate traffic between two groups of networks, enter:

```
ADD -Filter MASK any %0 | %ff = %ff
```

```
ADD -Filter POLicy wall Discard any BETWEEN !1, !2 AND !3, !4
```

Packets with any value at offset %0 meet the condition of mask any. Any packet received on port 1 or port 2 and sent to port 3 or port 4 is discarded, but packets received on port 1 and sent to port 2 are not discarded. Similarly, packets received on port 3 and sent to port 4, or packets that are received on port 4 and sent to port 3, are not discarded.

Example 14 If you want to discard all XNS broadcast packets, enter command A or command B. Command A is preferred because the built-in mask is encapsulation-independent.

Command A:

```
ADD -Filter POLicy noxnsbc Discard xns bc
```

Command B:

```
ADD -Filter MASK m1 %0 = %ffffffff
ADD -Filter MASK m2 %4 = %ffff
ADD -Filter MASK m3 %C = %0600
ADD -Filter POLicy p1 Discard m1, m2, m3
```

Table 103 explains the filter conditions in command B. All broadcast packets that have destination addresses of %ffffffff meet the conditions of the first and second masks. Only XNS packets meet the third condition.

Table 103 Filter Conditions

| | Mask m1 | Mask m2 | Mask m3 |
|----------|---|--|---|
| Offset | 0 | 4 | C |
| Meaning | First 4 bytes of destination address | Last 2 bytes of destination address | Packet type |
| Mask | ffffffff | ffff | 0600 |
| Operator | None | None | None |
| Effect | If first 4 bytes of destination address are ffffffff, the condition is met. | If last 2 bytes of destination address are ffff, the condition is met. | If packet is an XNS packet, the condition is met. |

Example 15 The following example shows the use of the logical OR operator. The following commands filter all packets that contain 500 (hexadecimal) or more bytes by applying the mask 11111111 to the byte at offset 500. If any value is present at that location, the filtering condition is met.

```
ADD -Filter MASK tail %500 | %ff = %ff
ADD -Filter POLicy drop Discard tail
```

Suppose the value 10110010 is present at offset 500 hexadecimal. When the logical OR operates on this value against the mask 11111111, the result is:

10110010 OR 11111111 = 11111111

Because the result is the same as the mask, the condition is met.

If no value is present at that location, the result is always false. Packets that contain more than 500 hexadecimal bytes should be blocked.

Applying a logical OR to any value and a mask of 11111111 always has a result of 11111111; if any value is present at byte 500, the condition is met. This means that any packet that contains 500 (hexadecimal) or more bytes is filtered.

Example 16 The following example shows the use of one logical operator:

```
ADD -Filter MASK andmask %a%80 = %80
```

```
ADD -Filter MASK ormask %a | %fe = %fe
ADD -Filter MASK notmask %a! = %8c
ADD -Filter POLicy together Discard andmask ormask notmask
```

In this example, all packets that meet the following three conditions are filtered:

Condition 1 This condition, %A:&%80, is met if the most significant bit of byte A is 1. It applies the logical AND operator to the value found at byte A and the mask 10000000. Suppose the value at byte A is 10111000:

```
          10111000
AND      10000000
          10000000
```

Because the result, 10000000, equals the mask, 10000000, the condition is met.

Condition 2 This condition, %A:\%FE, is met if the least significant bit of byte A is 0. It applies the logical OR operator to the value found at byte A and the mask 11111110. Suppose the value at byte A is 10111000:

```
          10111000
OR       11111110
          11111110
```

Because the result, 11111110, equals the mask, 11111110, the condition is met.

Condition 3 This condition, %A:!%8C, is met if byte A of the packet does not equal 8C. It compares the value found at byte A to the mask 10001100. Suppose the value at byte A is 10111000; because 10111000 is not equal to 10001100, this condition is met.

If a packet meets all three of these conditions, it is filtered. The packet used in this example meets all three conditions, because the value at byte A is assumed to be %B8; therefore, it is filtered.

A packet with the value 8F at byte A satisfies conditions 1 and 3, but does not meet condition 2; it is not filtered, but is forwarded to the appropriate destination.

Example 17 To add one specific address to the station group "accounting," enter:

```
add -Filter StationGroup accounting %080002123456
```

Example 18 To discard any traffic destined to the station group "accounting," enter:

```
add -Filter MASK to_accounting datalink.dstaddr = accounting
add -Filter POLicy block_account Discard to_accounting
```

Before entering these commands, enter the addresses of the stations belonging to the station group "accounting" using the ADD -Filter StationGroup command.

Example 19 To delete one specific address from the station group "accounting," enter:

```
DELeTe -Filter StationGroup accounting %080002123456
```

Example 20 To delete the station group "accounting," enter:

```
DELeTe -Filter StationGroup accounting
```



Before executing this command, you must delete all members of the station group "accounting" and delete any masks using the station group "accounting."

Example 21 To delete all members from the station group "accounting," enter:

```
DELeTe -Filter StationGroup accounting ALL
```

Example 22 To show the names of all station groups and the number of addresses in them, enter:

```
SHoW -Filter StationGroup
```

Example 23 To change the name of station group "bldg_100" to the station group "bldg_200," enter:

```
CHAnge -Filter StationGroup bldg_100 bldg_200
```

Example 24 This example illustrates how to allow NetWare Security Packets to go across a WAN dial-up link on port 4 only if the link is up, and be discarded if the link is down. You could set the WAN port to DOD and add a user-defined mask, NWSEC for the NetWare Security Packets. To add a filter policy for this, enter:

```
ADD -Filter POLIcy DROPNWSEC DODDISCARD NWSEC AT!4
```

Example 25 This example illustrates how to allow all broadcasts from port 1 to go across a WAN dial-up link on port 4 only if the link is up, and be discarded if the link is down. You could set the WAN port to DOD. You can then add a filter policy with a built-in mask, BC, by entering:

```
ADD -Filter POLIcy DROPBC DODDISCARD BC FROM !1 TO !4
```

Example 26 To create a mnemonic filter using the PROTOcolRsrv <tag> action option to allot 10 percent of the bandwidth to packets destined for a certain address that are passing through WAN port 3, follow these steps:

- 1 Add a filter mask with the name "DSTA_Mask" for a destination address of %0800AABB1111 by entering:

```
ADD -Filter MASK DSTA_MASK DL.DA = %0800AABB1111
```

- 2 Add a filter policy that will assign the name "dstpol" to the policy, select the name tag "dsta_tag" for the PROTOcolRsrv <tag> action option, and add the mask "dsta_mask" by entering:

```
ADD -Filter POLIcy dstpol PROTOcolRsrv DSTA_TAG DSTA_MASK
```

- 3 Enable the Filter Service by entering:

```
SETDefault -Filter CONTrol = Enable
```

- 4 Assign 10 percent of bandwidth to the PROTOcolRsrv name tag "dsta_tag" for port 3 by entering:

```
ADD !3 -PORT PROTOcolRsrv DSTA_TAG 10
```

- 5 Set PROTOcolRsrv as the -PORT QueueCONTrol parameter option for port 3 by entering:

```
SETDefault !3 -PORT QueueCONTrol = PROTOcolRsrv
```

After you have made these entries, any packet forwarded by the system matching the mask criteria is allotted 10 percent of the bandwidth in accordance with its name tag ("DSTA_TAG") and bandwidth allocation.

Example 27 This example shows how to use the PROTOcolRsrv <tag> action option to reserve a specified percentage of bandwidth for different protocols running on the same bridge/router.

In this example, in a bridge/router bridging IPX, XNS, and IP traffic, the user wants to reserve 40 percent of the bandwidth for IPX traffic, 35 percent for IP traffic, and 20 percent for XNS traffic, and 5 percent is set aside as a default for untagged traffic:

To allocate the required bandwidth for all the protocols, follow these steps:

- 1 Add a filter policy for each protocol with built-in IPX, IP, and XNS filter masks by entering:

```
ADD -Filter POLiCy POLICY1 PROTOcolRsrv ANY_IPX IPX
ADD -Filter POLiCy POLICY2 PROTOcolRsrv ANY_IP IP
ADD -Filter POLiCy POLICY3 PROTOcolRsrv ANY_XNS XNS
```

- 2 Select BRidging as the type of packet filtering to occur by entering:

```
SETDefault -Filter SElection = BRidging
```

- 3 Enable the Filter Service by entering:

```
SETDefault -Filter CONTrol = Enable
```

- 4 To define the bandwidth percentage to be reserved for each protocol, and to enter name tags that match those entered in the -Filter POLiCy commands, enter:

```
ADD !4 -PORT PROTOcolRsrv ANY_IPX 40
ADD !4 -PORT PROTOcolRsrv ANY_IP 35
ADD !4 -PORT PROTOcolRsrv ANY_XNS 20
```

- 5 Specify the PROTOcolRsrv option for the -PORT Service QueueCONTrol parameter by entering:

```
SETDefault !4 -PORT QueueCONTrol = PROTOcolRsrv
```

Example 28 This example shows how to use the PROTOcolRsrv <tag> action option to reserve a specified percentage of bandwidth for bridged packets of specified lengths being bridged outbound through a bridge/router WAN port.

In this example, in a bridge configured for IPX traffic, a user wants to reserve the following percentages of bandwidth for packets of the following lengths:

- 50 percent of the bandwidth for packets of a length less than 100 bytes
- 25 percent of the bandwidth for packets of a length between 100 and 400 bytes
- 20 percent of the bandwidth for packets of a length greater than 400 bytes
- 5 percent of the bandwidth is reserved as a default for untagged traffic.

To reserve the specified bandwidth for these packets, follow these steps:

- 1 Add a user-defined mask for each packet length condition that must be met by entering:

```
ADD -Filter MASK MYMASK1 IPX.PACKETLEN <100
ADD -Filter MASK MYMASK2 IPX.PACKETLEN 100-400
ADD -Filter MASK MYMASK3 IPX.PACKETLEN >400
```

- 2 Add filter policies to use the filter masks by entering:

```
ADD -Filter POLiCy POLICY_x PROTOcolRsrv MYTAG_A MYMASK1
```



```
ADD -Filter POLICY POLICY_y PROTOcolRsrv MYTAG_B MYMASK2
ADD -Filter POLICY POLICY_z PROTOcolRsrv MYTAG_C MYMASK3
```

- 3 Select BRidging as the type of packet filtering to occur by entering:

```
SETDefault -Filter SELECTION = BRidging
```

- 4 Enable the Filter Service by entering:

```
SETDefault -Filter CONTROL = Enable
```

- 5 Define the percentage of bandwidth to be reserved for each of the policies entered in step 4, and enter name tags that match those entered in step 4, by entering:

```
ADD !3 -PORT PROTOcolRsrv MYTAG_A 50
ADD !3 -PORT PROTOcolRsrv MYTAG_B 25
ADD !3 -PORT PROTOcolRsrv MYTAG_C 20
```

- 6 Specify the PROTOcolRsrv option for the -PORT Service QueueCONTROL parameter, by entering:

```
SETDefault !3 -PORT QueueCONTROL = PROTOcolRsrv
```

After you have made these entries, all IPX packets of lengths less than 100 bytes going outbound WAN port 3 get 50 percent of the bandwidth. Any IPX packets of a length between 100 and 400 bytes get 25 percent of the bandwidth, and IPX packets of a length greater than 400 bytes get 20 percent of the bandwidth.

Five percent of the bandwidth is reserved by default for untagged traffic. If the full 100 percent of bandwidth is allocated by the commands for various filtering conditions, the system normalizes the amount of bandwidth allotted for each condition so that there is always a reserve of 5 percent for untagged traffic.

IPX Filtering Examples

This section contains examples of filtering features in an IPX environment.

Setting Up IPX Filter Masks

The following examples illustrate how the mnemonic filter can be configured to set up filter masks in an IPX environment.

- Example 1* To create a mask named ipxmask1 that filters all IPX packets with the destination socket number equal to that of a NetWare Security Packet (0x457), enter:

```
ADD -Filter MASK ipxmask1 IPX.DstSocket = %0457
```

or

```
ADD -Filter MASK ipxmask1 IPX.DstSocket = NWSecPkt
```

- Example 2* To create a mask named ipxmask2 that filters all IPX packets with the destination network number 10 to 20, enter:

```
ADD -Filter MASK ipxmask2 IPX.DstNetwork 10-20
```

- Example 3* To create a mask named ipxmask3 that filters all IPX packets of length greater than 96, enter:

```
ADD -Filter MASK ipxmask3 IPX.PacketLength > 96
```

- Example 4* To create a mask named ipxmask4 that filters all IPX packets where the next 9 bytes match the string "MYSERVER1" (bytes starting from offset 4 bytes after the IPX header), enter:

```
ADD -Filter MASK ipxmask4 IPX.Data+%4:9 = "MYSERVER1"
```

Setting Up IPX Filter Policies

The following examples illustrate how the mnemonic filter can be configured to set up filter policies that manage IPX traffic in either a bridged or IPX routed environment. The examples assume that no other bridge or IPX policies are active except those that are explicitly configured in the examples. Bridge policies, if configured and selected, are always applied after the IPX policies have been applied and a no-match was the result.

Example 1 For a bridge, to discard all IPX packets with any socket number and forward all Service Advertising Protocol (SAP) and Routing Information Protocol (RIP) packets, follow these steps:

- 1 Create the user-defined IPX mask by entering:

```
ADD -Filter MASK IPXM1 ipx.SocKet = ALL
```

- 2 Create the filter policies by entering:

```
ADD -FI POLicy IPXP1 discard IPXM1
ADD -FI POLicy IPXP2 forward IPXRIP
ADD -FI POLicy IPXP3 forward SAP
```

The policies are applied as follows:

- An IPX packet is evaluated against policy IPXP2. If it matches, this packet (RIP) is forwarded. If it does not match (not a RIP packet), then it is evaluated against policy IPXP3.
- If it matches IPXP3, this packet (SAP) is forwarded. If it does not match, (not a SAP packet), then it is evaluated against policy IPXP1.

Since IPXP1 has a mask value of "socket = ALL," the packet matches and is discarded. A non-IPX packet is not subjected to those IPX policies, and the action taken depends upon the setting of the DefaultAction parameter. The default value of the DefaultAction parameter is Forward.

In general, an IPX policy using a user-defined IPX mask with the value of ALL is evaluated last among the list of IPX policies.



SAP and RIP packets are not subjected to IPX mnemonic filtering on an IPX router.

Example 2 For a bridge, to forward all SAP packets shorter than 100 bytes and discard all others, follow these steps:

- 1 Create the user-defined IPX mask by entering:

```
ADD -Filter MASK IPXM1 ipx.PacketLen < 100
```

- 2 Create the filter policies by entering:

```
ADD -FI POLicy IPXP1 discard SAP
ADD -FI POLicy IPXP2 forward SAP IPXM1
```

The policies are applied as follows:

- An IPX packet is evaluated against policy IPXP2. If it matches, then this packet (a SAP packet with IPX length less than 100 bytes) is forwarded. If it does not match, it is evaluated against policy IPXP1.
- If it matches policy IPXP1 (a SAP packet with IPX length equal or greater than 100 bytes) then this packet is discarded.
- If it matches none of the policies, then the action taken depends upon the setting of the DefaultAction parameter.

Policies that are more specific (with a greater number of masks or matching criteria) are applied ahead of less specific policies that have fewer matching criteria or masks. In this example, an IPX packet is evaluated against policy IPXP2 first, because IPXP2 uses a superset of the IPXP1 masks and is therefore more specific.

Example 3 To discard IPX packets from all clients except the client with the node address of %00608c37c0ba, follow these steps:

- 1 Set the DefaultAction parameter to forward by entering:

```
SETDefault -Filter DefaultAction = Forward
```

- 2 Create the user-defined IPX mask by entering:

```
ADD -Filter MASK IPXM1 ipx.SrcNodeAddr != %00608c37c0ba
```

- 3 Create the filter policies by entering:

```
ADD -Filter POLicy IPXP1 discard IPXM1
```

In this example, an IPX packet is evaluated against policy IPXP1. If it matches (an IPX packet that does not contain the source node address of %00608c37c0ba), then this packet is discarded. If it does not match, then the DefaultAction parameter is applied. In this example, the packet is forwarded.

Example 4 You can use a combination of policies, for example BRidge and IPX, to manage IPX traffic. To forward only IPX WAN Broadcast packets with the destination network of %45469220 and discard all other IPX packets, follow these steps:

- 1 Create a user-defined IPX mask by entering:

```
ADD -Filter MASK IPXM1 ipx.DstNETWORK = %45469220
```

- 2 Create the IPX filter policy by entering:

```
ADD -Filter POLicy IPXP1 forward WANBC IPXM1
```

- 3 Create the BRidge filter policy by entering:

```
ADD -Filter POLicy BRP1 discard IPX
```

The policies are applied as follows:

- An IPX packet is evaluated against policy IPXP1. If it matches, then this packet (a WANBroadcast packet containing the destination network of %45469220) is forwarded. If it does not match, then it is evaluated against BRidge policy BRP1.
- If it matches policy BRP1 (an IPX packet), then this packet is discarded.
- If the packet does not match any policies, the action taken depends upon the setting of the DefaultAction parameter.

Example 5 To discard NetWare security packets going out on a dial-on-demand port, enter:

```
ADD -Filter POLicy IPXP1 DodDiscard NWSEC
```

In this example, an IPX packet is evaluated against policy IPXP1. If the packet matches a NetWare security packet and is going out on a dial-on-demand port with its dial-up path down, the packet is discarded. If the dial-up path is up, the packet is forwarded but tagged so that it does not hold up the dial path. If a packet does not match, the action taken depends upon the setting of the DefaultAction parameter.

Example 6 This example illustrates how to count the number of IPX packets in each of the following IPX length categories:

Byte length of packets: <= 100
 > 100 and <= 200
 > 200 and <= 300
 > 300 and <= 400
 > 400

To create the masks and policies, follow these steps:

- 1 Create the user-defined IPX masks by entering:

```
ADD -Filter MASK IPXM1 ipx.PacketLen <= 100
ADD -Filter MASK IPXM2 ipx.PacketLen > 100
ADD -Filter MASK IPXM3 ipx.PacketLen <= 200
ADD -Filter MASK IPXM4 ipx.PacketLen > 200
ADD -Filter MASK IPXM5 ipx.PacketLen <= 300
ADD -Filter MASK IPXM6 ipx.PacketLen > 300
ADD -Filter MASK IPXM7 ipx.PacketLen <= 400
ADD -Filter MASK IPXM8 ipx.PacketLen > 400
```

- 2 Create the filter policies by entering:

```
ADD -Filter POLicy IPXP1 count IPXM1
ADD -Filter POLicy IPXP2 count IPXM2 IPXM3
ADD -Filter POLicy IPXP3 count IPXM4 IPXM5
ADD -Filter POLicy IPXP4 count IPXM6 IPXM7
ADD -Filter POLicy IPXP5 count IPXM8
```

In this example, an IPX packet is matched against IPXP1. If its length is less than 100 bytes, that count is incremented. If it does not match, then the packet is matched against IPXP2. If its length is greater than 100 but equal to or less than 200, that count is incremented. If it does not match, then it is matched against IPXP3. If its length is greater than 200 but equal to or less than 300, that count is incremented. If it does not match, then the packet is matched against IPXP4. If its length is greater than 300 but equal to or less than 400, that count is incremented. If it does not match, then the packet is matched against IPXP5. If its length is greater than 400, that count is incremented.

This example illustrates the use of multiple masks for the policies. See the next example (example 7) for an alternative configuration.

Example 7 This example illustrates a procedure for configuring IPX mnemonic filters to count various IPX packets by IPX length.

Byte length of packets: < 101
 101 - 200
 201 - 300
 301 - 400
 > 400

To create the masks and policies, follow these steps:

- 1 Create the user-defined IPX masks by entering:

```
ADD -Filter MASK IPXM1 ipx.PacketLen < 101
ADD -Filter MASK IPXM2 ipx.PacketLen 101 - 200
ADD -Filter MASK IPXM3 ipx.PacketLen 201 - 300
ADD -Filter MASK IPXM4 ipx.PacketLen 301 - 400
ADD -Filter MASK IPXM5 ipx.PacketLen > 400
```

- 2 Create the filter policies by entering:

```
ADD -Filter POLicy IPXP1 count IPXM1
ADD -Filter POLicy IPXP2 count IPXM2
ADD -Filter POLicy IPXP3 count IPXM3
ADD -Filter POLicy IPXP4 count IPXM4
ADD -Filter POLicy IPXP5 count IPXM5
```

In this example, an IPX packet is matched against IPXP1. If its length is less than 101 bytes, that count is incremented. If it does not match, then the packet is matched against IPXP2. If its length is between 101 and 200 inclusive, that count is incremented. If it does not match, then it is matched against IPXP3. If its length is between 201 and 300 inclusive, that count is incremented. If it does not match, then the packet is matched against IPXP4. If its length is between 301 and 400 inclusive, that count is incremented. If it does not match, then the packet is matched against IPXP5. If its length is greater than 400, that count is incremented.

Example 8 This example shows how to use mnemonic filtering to prioritize IPX traffic outbound on a WAN serial port 2. IPX packets are to be prioritized into high, medium, and low priorities according to their packet lengths. The following table shows the packet priority and IPX length:

| Priority | IPX Length |
|----------|-------------------|
| High | < 100 |
| Medium | >= 100 and <= 300 |
| Low | >300 |

To create the masks and policies, follow these steps:

- 1 Create the user-defined IPX masks by entering:

```
ADD -Filter MASK IPXM1 ipx.PacketLen < 100
ADD -Filter MASK IPXM2 ipx.PacketLen 100 - 300
ADD -Filter MASK IPXM3 ipx.PacketLen > 300
```

- 2 Create the filter policies by entering:

```
ADD -FI POLicy IPXP1 PRIOritize High IPXM1 to !2
ADD -FI POLicy IPXP2 PRIOritize Medium IPXM2 to !2
ADD -FI POLicy IPXP3 PRIOritize Low IPXM3 to !2
```

In this example, an IPX packet that matches IPXM1 (one that has an IPX length of less than 100 bytes) is placed into the high-priority output queue. An IPX packet that matches mask IPXM3 (one that has an IPX length greater than 300 bytes) is placed into the low-priority output queue. All other IPX packets match mask IPXM2 and are placed into the medium-priority output queue. The packets in the output queues are then sent out in a high:medium:low ratio that is configured using the -PORT QueueInterLeave parameter.

Example 9 To set up protocol reservation using the PROTOcolRsrv <tag> action option of the -Filter POLicy parameter so that all IPX packets greater than 400 bytes passing through WAN port number 4 get 25 percent of the bandwidth, follow these steps:

- 1 Add a user-defined mask called IPXMask that sets the following conditions for the passing packets: the packets must be IPX and the packet lengths must be greater than 400 bytes.

Enter:

```
ADD -Filter MASK IPXMask IPX.PACKETLEN > 400
```

- 2 Add a policy that includes the policy name IPXPolicy, the mask IPXMask, and the action option PROTOcolRsrv <tag>.

The PROTOcolRsrv <tag> action option includes entering a tag name IPXlarge to identify those packets that will receive the reserved bandwidth.

Enter:

```
ADD -Filter POLicy IPXPolicy PROTOcolRsrv IPXLARGE IPXMASK
```

- 3 Select BRIdging as the type of packet filtering to occur by entering:

```
SETDefault -Filter SELECTION = BRIdging
```

- 4 Enable the Filter Service by entering:

```
SETDefault -Filter CONTROL = Enable
```

- 5 Add the same physical port and the same tag name as was entered in the Filter Service POLicy command. Also, enter the 25 percent of bandwidth to be reserved for the designated protocol name.

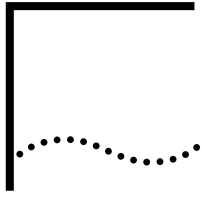
Enter:

```
ADD !4 -PORT PROTOcolRsrv IPXLARGE 25
```

- 6 Specify the PROTOcolRsrv option for the -PORT QueueCONTROL parameter by entering:

```
SETDefault !4 -PORT QueueCONTROL = PROTOcolRsrv
```

After this configuration, if the system forwards a packet that contains a matching FILTER POLicy, the system provides a queue with the percentage of bandwidth reserved for this PROTOcolRsrv <tag>.



CONFIGURING PROTOCOL RESERVATION

The protocol reservation feature enables you to assign a percentage of bandwidth to designated packets transmitting out of a WAN port that meet certain conditions.

This chapter describes how to configure protocol reservation for the following bridged- and routed-protocol packet types:

- IP-routed packets
- IPX-routed packets and all bridged packets
- NETBuilder-supported IBM traffic types, including DLSw (endpoint), LLC2 (for both SNA and NetBIOS) and APPN-routed packets



The reservation of bandwidth for packets transmitting over X.25 is not supported by protocol reservation. See the [Configuring Wide Area Networking Using X.25](#) chapter for bandwidth management solutions for packets using X.25.

Protocol reservation only affects traffic being transmitted from the local bridge/router (the transmit direction). You cannot configure protocol reservation for traffic being received by the local bridge/router (the receive direction). If you want protocol reservation for traffic in both directions, then you must configure protocol reservation on both bridge/routers (the local and remote) that are sending traffic to each other.

Protocol reservation can be set for all WAN ports on the bridge/router or for specified WAN ports. However, configuring protocol reservation for specific ports is not recommended because it can affect network performance.



For conceptual information, see “[How Protocol Reservation Works](#)” later in this chapter.

Why Use Protocol Reservation

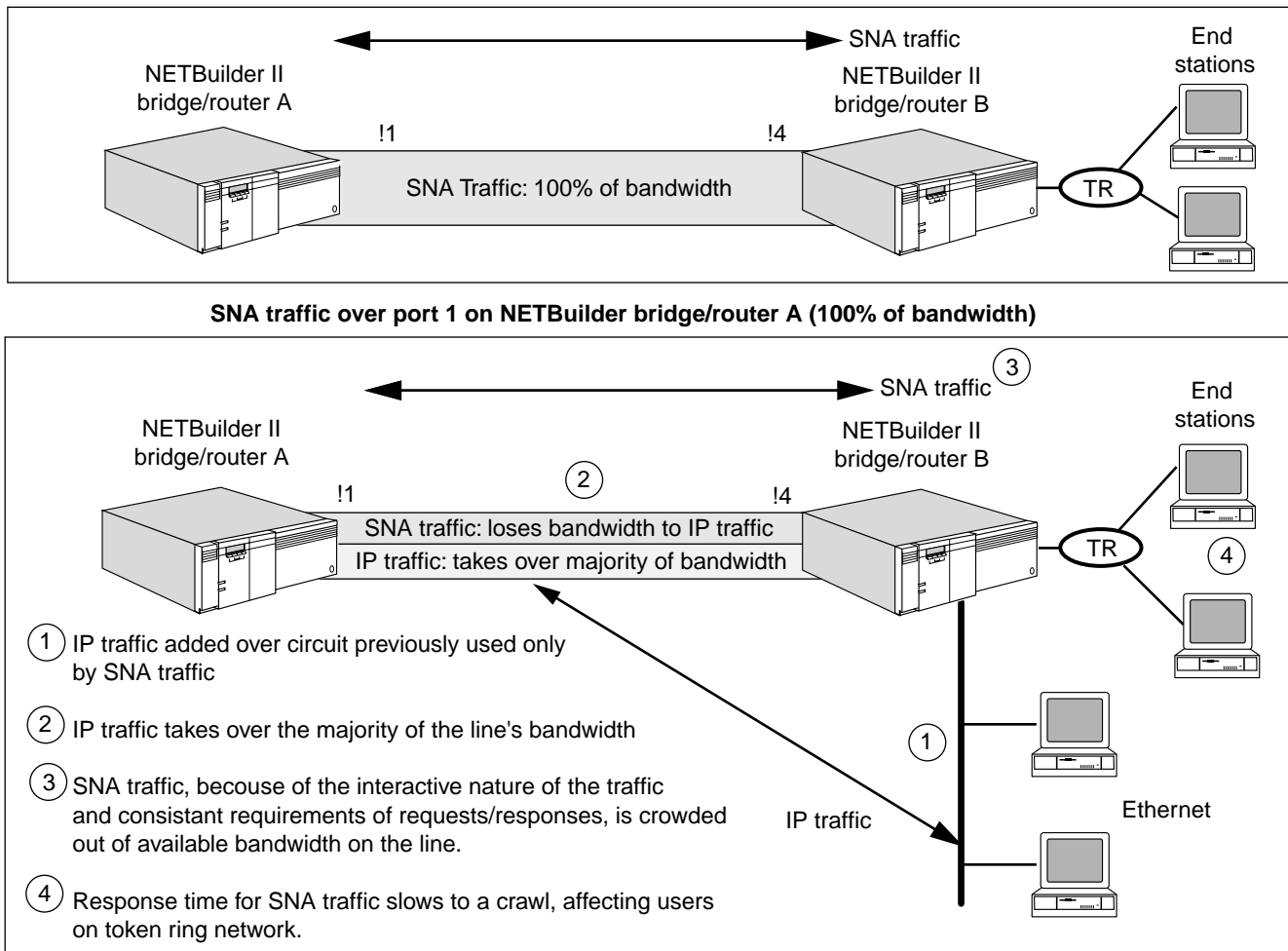
Protocol reservation enables you to reserve bandwidth for lower bandwidth usage, interactive, response-time-sensitive, or transaction-oriented network applications, which are normally crowded out by heavy bandwidth usage applications such as file transfer or mail.

For example, in a multiprotocol environment that includes IBM protocol traffic (such as response-time-sensitive and mission-critical SNA packets) mixed in with other protocol traffic (such as IP or IPX), SNA devices throttle back the data transmission rate to the end station when they sense available bandwidth decreasing. If other network protocols increase this bandwidth consumption, SNA devices will throttle back the data transmission rate more, which slows the response time of SNA packets even more.

To avoid this situation, use protocol reservation to provide a percentage of bandwidth for the SNA packets and to restrict the percentage of bandwidth to the other more aggressive protocol packets. This will ensure that the small, response-time-sensitive SNA packets can pass through the port in a timely manner.

Figure 439 shows this situation. In the first diagram, only interactive SNA traffic is travelling over the port, using up the available bandwidth that provides the end user adequate response time. However, if you decide to also send IP packet traffic over that same port (as shown in the second diagram) then the IP packets continually use as much bandwidth as possible until the SNA traffic is "crowded out" of the bandwidth, which greatly reduces the response time of the SNA devices on the network. This crowding out is due to the connectionless nature of how IP works versus the connection-oriented nature of SNA interactive traffic.

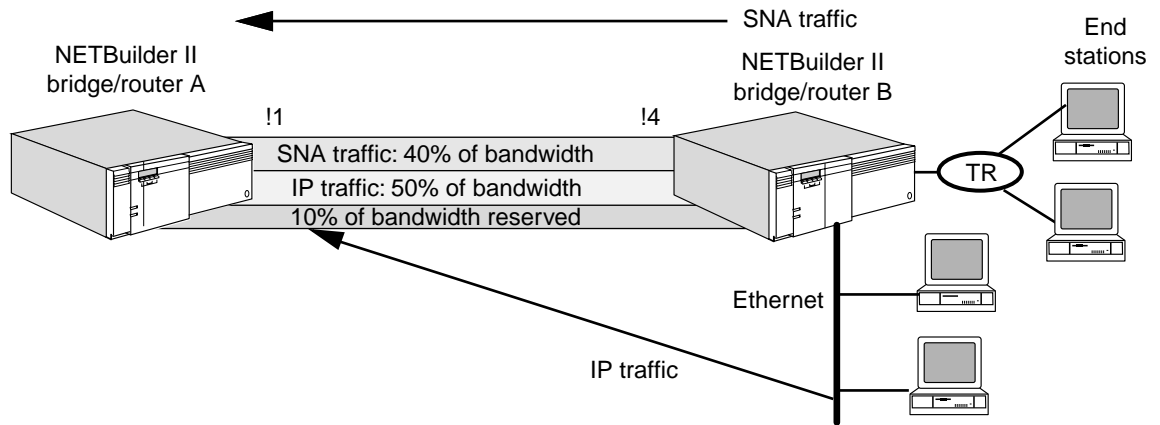
Figure 439 IP and SNA Traffic Contention (Without Protocol Reservation)



To deal with this situation, you can use protocol reservation to reserve a percentage of the port bandwidth to each protocol. Figure 440 shows this same example, only with IP allocated 50 percent of bandwidth and SNA traffic allocated 40 percent of traffic (the other 10 percent is the default, for other traffic).

The allocation of the bandwidth configured with protocol reservation occurs only when the different packet types actually contend for the bandwidth of the configured port.

Figure 440 IP and SNA Traffic Contention (With Protocol Reservation)



SNA traffic and IP packets dividing percentage of bandwidth based on protocol reservation percentages

Protocol reservation can be used to allocate recommended bandwidth for other protocols besides IP and SNA traffic. You also have wide flexibility in determining which protocols you want to reserve bandwidth to, and how much. For more information about how protocol reservation works, see "How Protocol Reservation Works" later in this chapter.

Protocol Reservation Procedural Overview

This section provides an overview of how to configure protocol reservation for different traffic types. Because procedures for each of the traffic types varies, read this section to determine the proper procedure for your configuration.

For specific step-by-step configuration procedures, see "Configuring for Bridged Traffic or IP- or IPX-Routed Traffic" or "Configuring for IBM Traffic" later in this chapter.

Protocol reservation can be configured using a variety of procedures, depending on the type of packet traffic you are configured. Table 104 lists the traffic types that can be configured for protocol reservation and the procedure used to configure each. More detailed information about the procedures for each traffic type follows the table.

Table 104 Packet Types and Configuration Procedures for Protocol Reservation

| Traffic Types | Configuration Procedure | Mask | Tag | See Configuration Examples in This Chapter |
|---|-------------------------|---|--------------|--|
| All Bridged traffic including IP, IPX, AppleTalk, XNS, SNA, NetBIOS | Filter Service* | Built-in masks or user-defined masks
SNA and NetBIOS need user-defined bridged masks | User-defined | "Configuring for Bridged Traffic" |

Table 104 Packet Types and Configuration Procedures for Protocol Reservation (continued)

| Traffic Types | Configuration Procedure | Mask | Tag | See Configuration Examples in This Chapter |
|--|---|--------------------------------------|-----------------------|---|
| IP-Routed traffic such as DLSw (<i>within</i> DLSw tunnel), FTP, IP, IPDATA, ICMP, SMTP, TCP, TELNET, and UDP | -IP FilterAddr parameter [†] | Built-in protocol masks | User-defined | "Configuring for IP-Routed Packets" |
| IPX-routed traffic | Filter Service* | Built-in masks or user-defined masks | User-defined | "Configuring for IPX-Routed Traffic" |
| DLSw traffic for a port on a bridge/router that is the endpoint of the DLSw tunnel | -PORT PROTOcolRsrv parameter [‡] | Not applicable | Built-in DLSW tag | "Configuring for DLSw Traffic at the Tunnel Endpoint" |
| DLSw traffic to a specific DLSw peer that is the endpoint of the DLSw tunnel | -PORT PROTOcolRsrv parameter** | Not applicable | Built-in DLSWPEER tag | "Configuring for DLSw Traffic at the Tunnel Endpoint" |
| LLC2 traffic, which carries SNA and NetBIOS packets | -Filter Service* | Built-in LLC2 masks: SNA, or NetBIOS | User-defined | "Configuring for LLC2 Traffic for SNA Boundary Routing" |
| APPN-routed traffic | -Filter Service* | Built-in LLC2 mask: APPN | User-defined | "Configuring for APPN-Routed Traffic" |

* See the Configuring Mnemonic Filtering chapter in this guide and the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

† See "Configuring Packet Filtering" in the Configuring IP Routing chapter for IP filtering examples, and to the IP Service Parameters chapter in *Reference for Enterprise OS Software* for parameter syntax.

‡ See the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

**See the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

When you configure a Filter POLicy for use with protocol reservation, the Filter POLicy should not specify a port number. Not specifying a port number will ensure that the protocol reservation valve will control bandwidth as defined. The examples in this chapter follow this recommendation and will operate correctly regardless of the configuration of the WAN port (Frame Relay with or without virtual ports, PPP, or WAN Extender).

Protocol reservation uses bandwidth allocation rules to determine how to allocate bandwidth of one traffic type compared to the bandwidth of other traffic types. For more information, see "Bandwidth Allocation Process Rules" later in this chapter.



When you enter the -PORT PROTOcolRsrv command, you must specify a physical WAN port, not a virtual port. This rule applies for all bridge/router port configurations with the exception of WANExtender ports, where you must enter the -PORT PROTOcolRsrv command and specify a virtual port.

More specifically, protocol reservation is configured using the following procedures for each traffic type:

- Procedure for mnemonic filtering

To configure protocol reservation using the mnemonic filtering procedure, perform the following major tasks:

- Using various Filter Service parameters, assign a built-in or user-defined mask, create a filter policy, and designate the type of packet filtering that is being performed.

See the Configuring Mnemonic Filtering chapter for mnemonic filtering descriptions and lists of built-in masks and instructions on how to create user-defined masks. See the Filter Service Parameters chapter in *Reference for Enterprise OS Software* for syntax and descriptions of Filter Service parameters.

- Set the -PORT QueueCONTROL parameter to PROTOcolRsrv and use the -PORT PROTOcolRsrv parameter to assign bandwidth percentage and a tag for the packet traffic type.

See the PORT Service Parameters chapter in *Reference for Enterprise OS Software* for syntax and descriptions of the -PORT PROTOcolRsrv parameter.

- Procedure for IP filtering

To configure protocol reservation using the IP filtering procedure, perform the following major tasks:

- Use IP Service parameters to create a filter and enable filtering.

See “Configuring Packet Filtering” in the Configuring IP Routing chapter for IP filtering descriptions and examples. See the IP Service Parameters chapter in *Reference for Enterprise OS Software* for syntax and descriptions of the IP Service parameters.

- Set the -PORT QueueCONTROL parameter to PROTOcolRsrv, and use the -PORT PROTOcolRsrv parameter to assign bandwidth percentage and a tag for the packet traffic type.

See the PORT Service Parameters chapter in *Reference for Enterprise OS Software* for syntax and descriptions of the -PORT PROTOcolRsrv parameter.

- Procedure for DLSw

To configure a port for protocol reservation using the DLSw procedure (used for all DLSW tunnel endpoint packets or packets designated for a DLSW peer for an end of the DLSw tunnel — traffic that will not be routed forward), perform the following major tasks:

- Set the -PORT QueueCONTROL parameter to PROTOcolRsrv.
- Select either the DLSw tag or the DLSWPEER tag (and enter the peer's IP address) from the -PORT PROTOcolRsrv parameter options, and enter the percentage of bandwidth to be designated for the DLSw or DLSWPEER packet type.

See the PORT Service Parameters chapter in *Reference for Enterprise OS Software* for the syntax and descriptions of the -PORT QueueCONTROL and -PORT PROTOcolRsrv parameters.

Using Protocol Reservation with Frame Relay Virtual Ports

When you configure protocol reservation for traffic being sent over Frame Relay virtual ports, you must configure protocol reservation on the physical port. You can set up the filter to tag packets on virtual ports, but if you configure the filters using this method, you must configure filters for *all* virtual ports assigned to the physical port. If you have a large number of virtual ports and you configure protocol reservation filters for each virtual port, system performance will be negatively impacted. 3Com recommends that you configure the protocol

reservation filters to apply to the bridge/router instead of individual ports, then configure the protocol reservation percentages to apply to individual ports.

Configuring for Bridged Traffic or IP- or IPX-Routed Traffic

This section describes how to configure protocol reservation for IP-routed packets, bridged traffic, and IPX-routed packets. This section provides the following procedures:

- Configuring for Bridged Traffic
- Configuring for IP-Routed Packets
- Configuring for IPX-Routed Traffic



The procedures in this section only describe how to configure a single traffic type at one time. For configuration examples on how to configure mixed environments, see "Protocol Reservation Configuration Examples" later in this chapter.

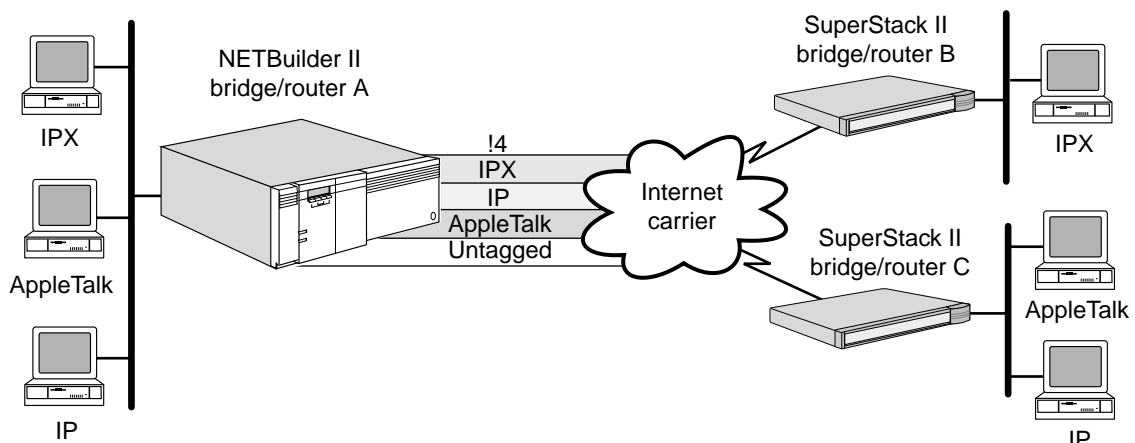
Configuring for Bridged Traffic

This section provides an example on how to use the mnemonic filtering procedure to configure protocol reservation for all bridged protocol packets such as IP, IPX, and AppleTalk.

In this example, a bridge/router is bridging IPX, IP, and AppleTalk traffic. The user wants to reserve 40 percent of the bandwidth for IPX traffic, 35 percent for IP traffic, and 20 percent for AppleTalk traffic transmitting from WAN port 4.

Five percent of the bandwidth is automatically set aside as a default for untagged traffic. Figure 441 illustrates this example.

Figure 441 Hardware Configuration for Bridged Packets Example



To allocate the required bandwidth for these bridged protocols, follow these steps on NETBuilder II bridge/router A:

- 1 Add a filter policy for each protocol with built-in IPX, IP, and AppleTalk filter masks by entering:

```
ADD -Filter POLicy POLICY1 PROTOcolRsrv ANY_IPX IPX
ADD -Filter POLicy POLICY2 PROTOcolRsrv ANY_IP IP
ADD -Filter POLicy POLICY3 PROTOcolRsrv ANY_APPLE ATALK
```



When you configure filters to control protocol reservation, the filters will affect all ports on the bridge/router. You can configure filters for specific ports, but this configuration is not recommended for use with protocol reservation.

- 2 Apply the bridge filtering policies by entering:

```
SETDefault -Filter SElection = BRidge
```

- 3 Enable the Filter Service by entering:

```
SETDefault -Filter CONTROL = Enable
```

- 4 Define the percentage of bandwidth to be reserved for each protocol and enter name tags that match those entered in the -Filter POLicy commands in step 1 by entering:

```
ADD !4 -PORT PROTOcolRsrv ANY_IPX 40
ADD !4 -PORT PROTOcolRsrv ANY_IP 35
ADD !4 -PORT PROTOcolRsrv ANY_APPLE 20
```



If configuring protocol reservation on a WAN Extender port, enter the PROTOcolRsrv command specifying a virtual port instead of a physical port.

- 5 Specify the PROTOcolRsrv option for the -PORT Service QueueCONTROL parameter by entering:

```
SETDefault !4 -PORT QueueCONTROL = PROTOcolRsrv
```

For more information and examples on how to use the mnemonic filtering procedure to set up protocol reservation, see the Configuring Mnemonic Filtering chapter.

Configuring for IP-Routed Packets

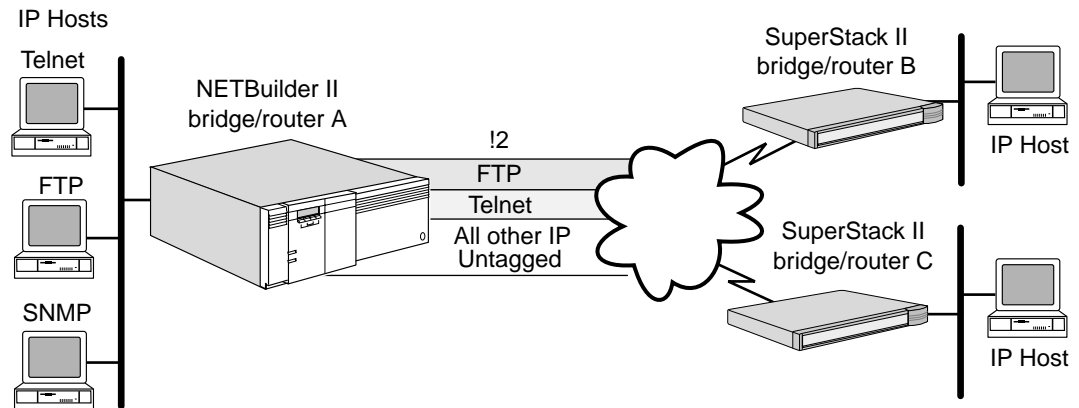
This section provides an example on how to use the IP filtering procedure to configure protocol reservation for IP-routed packets.

Prerequisites

Before beginning this procedure, complete the following tasks on the NETBuilder II bridge/router:

- Add an IP filter that assigns 20 percent of reserved bandwidth for all Telnet sessions, 30 percent of reserved bandwidth for all FTP packets, and 25 percent for all other IP packets transmitted through port 2.
- Set the -IP FilterDefAction parameter so that all packets that do not meet the filtering conditions are forwarded. Figure 442 shows the hardware configuration for this example.

Figure 442 Hardware Configuration for IP-Routed Packets Example



Procedure

To configure these filtering operations, follow these steps on NETBuilder II bridge/router A:

- 1 Set up IP routing according to the information in the Configuring IP Routing chapter.
- 2 Add IP filters that do the following for packets:
 - Assign 20 percent of reserved bandwidth for all Telnet packets and designate a tag name of "Telnet" to identify the packets.
 - Assign 30 percent of reserved bandwidth for all FTP packets and designate a tag name of "FTP" to identify the packets.
 - Assign 25 percent of reserved bandwidth for all other IP packets and designate a tag name of "ALLOther-IP" to identify the packets.

Add these filters by entering:

```
ADD -IP FilterAddr ALL ALL PROTOcolRsrv = TELNETTAG Telnet
ADD -IP FilterAddr ALL ALL PROTOcolRsrv = FTPTAG FTP
ADD -IP FilterAddr ALL ALL PROTOcolRsrv = ALLOther-IP IP
```



When you configure filters to control protocol reservation, the filters will affect all ports on the bridge/router. You can configure filters for specific ports, but this configuration is not recommended for use with protocol reservation.

- 3 Add an IP filter default action that forwards any packets that do not satisfy the filter requirements by entering the following command:

```
SETDefault -IP FilterDefAction = Forward
```

- 4 Enable the IP filtering feature by entering:

```
SETDefault -IP CONTROL = Filtering
```

- 5 Assign 20 percent of bandwidth to the PROTOcolRsrv name tag "Telnet," 30 percent of the bandwidth to the PROTOcolRsrv name tag "FTP," and 25 percent of bandwidth to the PROTOcolRsrv name tag "ALLOther-IP" for port 2 by entering:

```
ADD !2 -PORT PROTOcolRsrv TELNETTAG 20
ADD !2 -PORT PROTOcolRsrv FTPTAG 30
ADD !2 -PORT PROTOcolRsrv ALLOther-IP 25
```



If you are configuring protocol reservation on a WAN Extender port, enter the PROTOcolRsrv command specifying a virtual port instead of a physical port.

- 6 Set PROTOcolRsrv as the QueueCONTRol option for port 2 by entering:

```
SETDefault !2 -PORT QueueCONTRol = PROTOcolRsrv
```

After you have entered these commands, any packet sent out by the system through port 2 that has the name tag "Telnet" is allocated 20 percent of the bandwidth, packets with the name tag "FTP" are allocated 30 percent of the bandwidth, and IP packets with the name tag "ALLOther-IP" are allocated 25 percent.

Five percent of the bandwidth is allocated for all untagged traffic, and the remaining 20 percent of the bandwidth is added to the default to be used by the configured protocols or by the untagged traffic on a first-come first-serve basis.

For more information and examples on how to use the IP filtering procedure to set up protocol reservation, see "Configuring Packet Filtering" in the Configuring IP Routing chapter.

How Protocol Reservation Allocates Different IP Protocol Types

Using IP filtering, how you define the tags for IP packets or other protocols in the TCP/IP protocol suite determines how much percentage bandwidth is used for each. If you configure a percentage of bandwidth for a specific protocol, such as UDP, TCP, or Telnet, then those packets will be removed from the percentage allocated to IP. However, if you define a percentage of bandwidth for IP only, then all the IP-related protocols such as UDP, TCP, and Telnet will be included within that percentage.

Figure 443 is an example of how this allocation works. In the example, 60 percent of the bandwidth is allocated to IP.

Figure 443 IP Protocols Allocation (UDP Included)

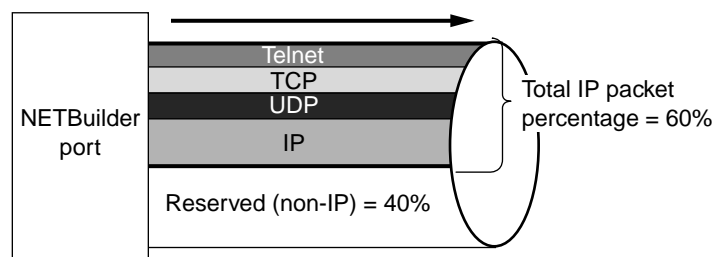
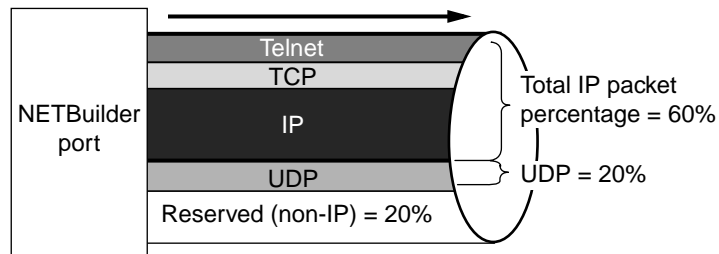


Figure 444 shows the same situation but with 60 percentage of bandwidth allocated to IP and 20 percent allocated to UDP. While UDP traffic is no longer included in the 60 percent of bandwidth allocated to IP, TCP and Telnet are still

allocated as a subset of the IP bandwidth percentage. The bandwidth allocated to UDP in this case is exclusive of bandwidth allocated to IP.

Figure 444 IP Protocols Allocation (UDP Excluded)



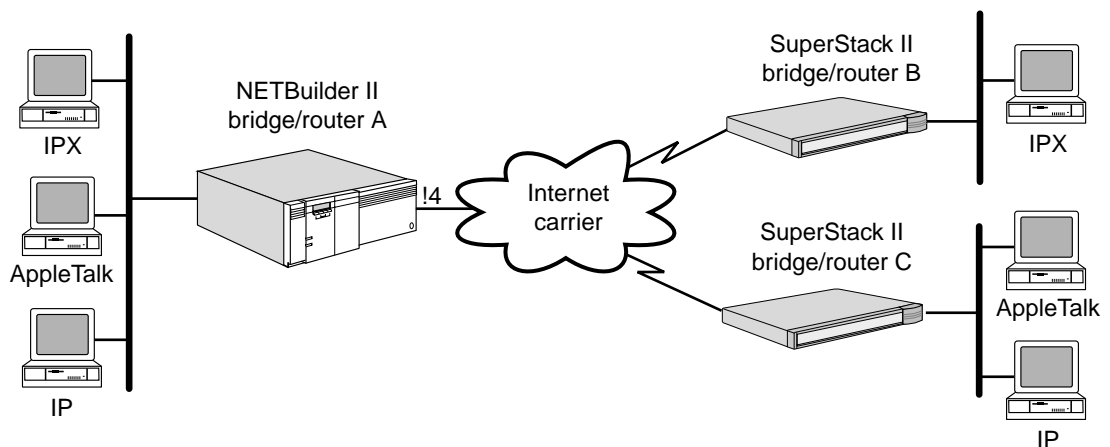
Configuring for IPX-Routed Traffic

The following example describes how to configure protocol reservation for IPX-routed traffic (IPXRIP packets) transmitted from WAN port 4 on a central node NETBuilder bridge/router to an end node NETBuilder bridge/router.

In this example, WAN port 4 on bridge router A is configured for protocol reservation to reserve the following bandwidth percentages for the following packet types (see Figure 445):

- 45 percent of the bandwidth for IPXRIP-routed packets
- 50 percent of the bandwidth for IP-routed packets
- 5 percent as a default for AppleTalk-routed packets

Figure 445 Hardware Configuration for IPX-Routed Traffic



To allocate the required bandwidth for these protocols, follow these steps on NETBuilder II bridge/router A:

- 1 Set up IPX routing according to the information in the Configuring IPX Routing chapter, set up IP routing according to the Configuring IP Routing chapter, and AppleTalk routing according to the Configuring AppleTalk Routing chapter.
- 2 Add a filter policy named "IPXPolicy" with PROTOcolRsrv as the action option, with the name tag "IPXtag" and with the built-in mask "IPXRIP" by entering:

```
ADD -Filter POLICY IPXPOLICY PROTOcolRsrv IPXTAG IPXRIP
```




When you configure filters to control protocol reservation, the filters will affect all ports on the bridge/router. You can configure filters for specific ports, but this configuration is not recommended for use with protocol reservation.

- 3 Apply the filtering policy by entering:
SETDefault -Filter SElection = IPX
- 4 Set the Filter CONTrol parameter to Enable by entering:
SETDefault -Filter CONTrol = Enable
- 5 Add an IP filter that assigns all IP-routed traffic the name tag "IPtag" by entering the following command:
ADD -IP FilterAddrs ALL ALL PROTOcolRsrv = IPTAG IP
- 6 Add an IP filter default action that forwards any packets that do not satisfy the filter requirements by entering:
SETDefault -IP FilterDefAction = Forward
- 7 Enable the IP filtering feature by entering:
SETDefault -IP CONTrol = Filtering
- 8 Configure port 4 with the "IPXtag" PROTOcolRsrv name tag entered in the filter policy with 45 percent of reserved bandwidth, and with the "IPtag" entered in the IP filter with 50 percent of the bandwidth by entering:
ADD !4 -PORT PROTOcolRsrv IPXTAG 45
ADD !4 -PORT PROTOcolRsrv IPTAG 50



If configuring protocol reservation on a WAN Extender port, enter the PROTOcolRsrv command specifying a virtual port instead of a physical port.

- 9 Set PROTOcolRsrv as the -PORT QueueCONTrol parameter option for port 4 by entering:
SETDefault !4 -PORT QueueCONTrol = PROTOcolRsrv

After this configuration, 45 percent of the port 4 bandwidth is reserved for IPXRIP packets, 50 percent for IP-routed traffic, and 5 percent is the default untagged traffic, which is AppleTalk in this example.

Configuring for IBM Traffic

This section describes how to configure protocol reservation for IBM traffic types. This section provides instructions for the following procedures:

- Configuring for DLSw Traffic at the Tunnel Endpoint
- Configuring for LLC2 Traffic for SNA Boundary Routing
- Configuring for APPN-Routed Traffic



The procedures in this section only describe how to configure a single traffic type at one time. For configuration examples on how to configure mixed traffic environments, see "Protocol Reservation Configuration Examples" later in this chapter.

The following IBM traffic supported by the NETBuilder bridge/routers can be configured for protocol reservation: DLSw, APPN-routed, LLC2 locally terminated (by DLSw or LLC2 tunneling), and SNA and NetBIOS, bridged traffic.

DLSw traffic is used to encapsulate SNA or NetBIOS traffic that is transmitting over a WAN. The DLSw traffic is itself encapsulated in IP traffic frames.

How DLSw traffic is configured for protocol reservation depends on whether the DLSw traffic is being configured for a bridge/router that is the end of the DLSw tunnel or if the DLSw traffic is to be forwarded on through the tunnel to another bridge/router.

Configuring protocol reservation for DLSw traffic for a bridge/router that is the end of the DLSw tunnel is accomplished using the DLSw or DLSwPeer built-in tags. You configure the -PORT PROTOcolRsrv parameter and enter DLSw as the name tag option and enter the percentage of bandwidth to be reserved. For more information, see “Configuring for DLSw Traffic at the Tunnel Endpoint” next.

Configuring a bridge/router for protocol reservation that is forwarding DLSw traffic through the DLSw tunnel is accomplished using the IP filtering procedure that uses IP Service parameters. See “Configuring for IP-Routed Packets” earlier in this chapter for instructions on how to use the IP filtering procedure; substitute the DLSw built-in tag for the TELNET or FTP built-in tag in the example.

The APPN built-in mask is used for IBM APPN-routed packets. See “Configuring for APPN-Routed Traffic” later in this chapter.

The following built-in masks are provided for IBM LLC2 traffic at a WAN port where DLSw or LLC2 tunneling is locally terminating the LLC2 connection, for example, at the WAN port connecting a NETBuilder boundary router central site with a NETBuilder leaf node:

- SNA – Used as criteria to select SNA traffic packets.
- NetBIOS – Used as criteria to select NetBIOS traffic packets.

These masks are used as criteria to select the packets to be the recipients of the reserved bandwidth. The packets are identified by name tags that are entered with the -PORT PROTOcolRsrv parameter. See “Configuring for LLC2 Traffic for SNA Boundary Routing” later in this chapter.

Configuring for DLSw Traffic at the Tunnel Endpoint

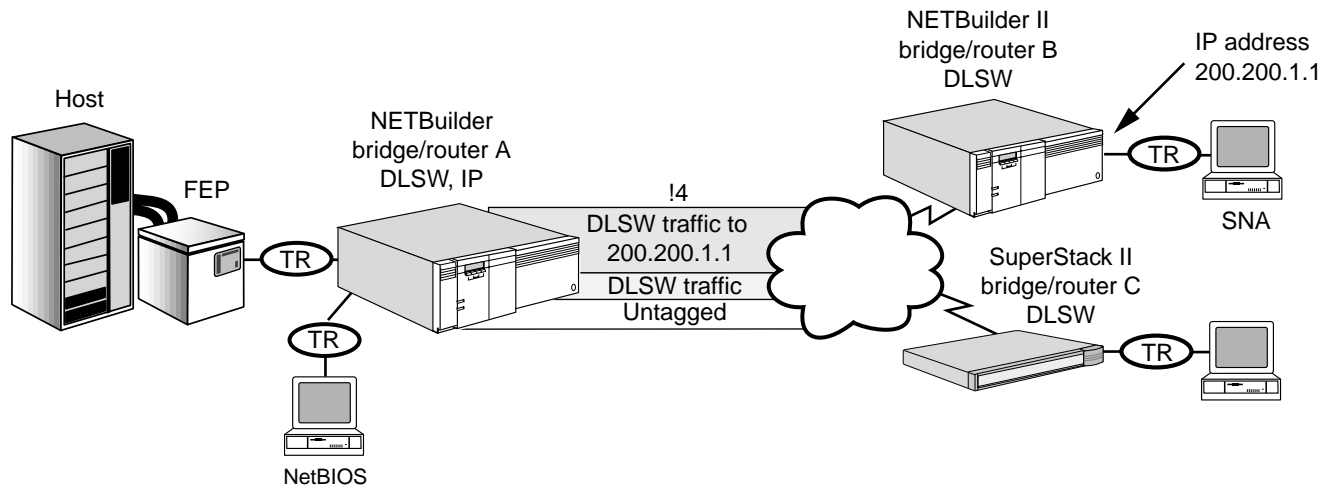
The following example describes how to configure protocol reservation to assign a percentage of a WAN port's bandwidth for DLSw traffic transmitting from a port on a bridge/router that is the endpoint of a DLSw tunnel. The DLSw packets (which are encapsulated within IP packets) can carry SNA packets, NetBIOS packets, or both.

This example allocates bandwidth of a DLSw tunnel endpoint on port 4 as follows:

- 70 percent of the bandwidth for all DLSw traffic
- 25 percent of the bandwidth for DLSw traffic destined for the DLSw peer at IP address 200.200.1.1

Figure 446 shows the hardware configuration for this example.

Figure 446 DLSw Tunnel Endpoint Hardware Configuration



To allocate the required bandwidth for the DLSw traffic in this example, follow these steps:

- 1 Set up a DLSw tunnel according to the information in the Configuring Data Link Switching for SNA and NetBIOS Networks chapter.
- 2 Assign 70 percent of bandwidth for DLSw traffic and 25 percent of the bandwidth for traffic destined for the DLSw peer at IP address 200.200.1.1 on port 4 by entering (DLSw and DLSwPeer have a built-in tags so you do not need to enter a tag):

```
ADD !4 -PORT PROTOcolRsrv DLSw 70
ADD !4 -PORT PROTOcolRsrv DLSwPeer 200.200.1.1 25
```



If configuring protocol reservation on a WAN Extender port, enter the PROTOcolRsrv command specifying a virtual port instead of a physical port.

- 3 Set PROTOcolRsrv as the -PORT QueueCONTRol parameter option for port 4 by entering:

```
SETDefault !4 -PORT QueueCONTRol = PROTOcolRsrv
```

After you have completed this configuration, 70 percent of the bandwidth is reserved for all DLSw traffic transmitting out of port 4, and 25 percent of the bandwidth is reserved for the DLSw peer with IP address 200.200.1.1. The 5 percent of bandwidth is the default to be used for untagged traffic transmitting from port 4.

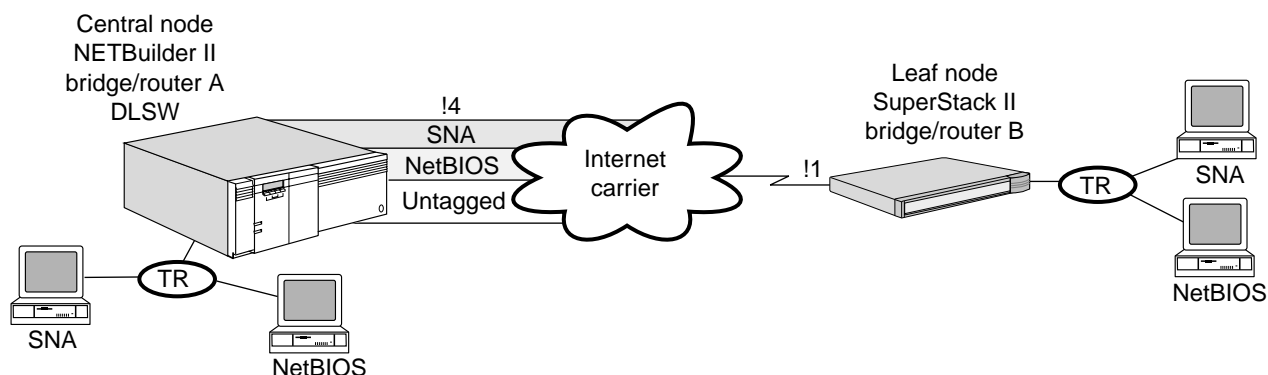
Configuring for LLC2 Traffic for SNA Boundary Routing

The following example describes how to set up protocol reservation for LLC2 traffic (carrying SNA or NetBIOS packets, or both) transmitting from a WAN port on a NETBuilder bridge/router serving as a central node to another NETBuilder bridge/router serving as an end node.

In this example, WAN port 4 on the central node bridge/router is configured for protocol reservation to reserve the following bandwidth percentages for the following packet types (see Figure 447):

- 50 percent of the bandwidth for SNA-bridged packets
- 45 percent of the bandwidth for NetBIOS-bridged packets

Figure 447 LLC2 Example Hardware Configuration



To configure protocol reservation for SNA and NetBIOS packets encapsulated in LLC2 traffic, follow these steps:

- 1 Configure bridging according to the Configuring Bridging chapter.
- 2 Assign the following filter policies:
 - A policy named "SNAPolicy," with the built-in mask SNA, and with PROTOcolRsrv as the action option and the name tag "SNAtag" with no port number specified.
 - A policy named "NetBIOSPol," with the built-in mask NetBIOS, and with PROTOcolRsrv as the action option and the name tag "NetBIOStag."

Assign these policies by entering:

```
ADD -Filter POLicy SNAPOLICY PROTOcolRsrv SNATAG SNA
ADD -Filter POLicy NETBIOSPOL PROTOcolRsrv NETBIOSSTAG NetBIOS
```



When you configure filters to control protocol reservation, the filters will affect all ports on the bridge/router. You can configure filters for specific ports, but this configuration is not recommended for use with protocol reservation.

- 3 Apply the filtering policies by entering:


```
SETDefault -Filter SElection = LLC2
```
- 4 Set the Filter CONTROL parameter to Enable by entering:


```
SETDefault -Filter CONTROL = Enable
```
- 5 Using the -PORT PROTOcolRsrv parameter, configure the WAN port 4 with the name tags assigned in step 2 for the following bandwidth percentages:
 - SNAtag 25 percent
 - NetBIOStag 20 percent

Assign these percentages by entering:

```
ADD !4 -PORT PROTOcolRsrv SNATAG 50
ADD !4 -PORT PROTOcolRsrv NETBIOSSTAG 45
```



If configuring protocol reservation on a WAN Extender port, enter the `PROTOCOLRsrv` command specifying a virtual port instead of a physical port.

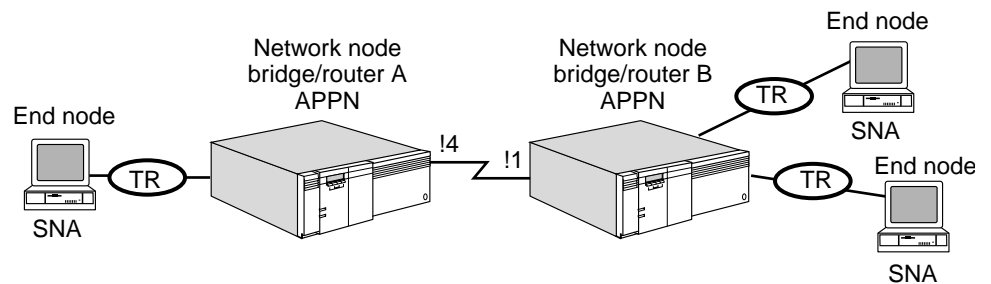
- Set the `-PORT QueueCONTROL` parameter to `PROTOCOLRsrv` for port 4 by entering:

```
SETDefault !4 -PORT QueueCONTROL = PROTOCOLRsrv
```

Configuring for APPN-Routed Traffic

The following example describes how to configure protocol reservation so that APPN-routed traffic transmitted from WAN port 4 on a central node NETBuilder bridge/router to an end node NETBuilder bridge/router is assigned 10 percent of the bandwidth (see Figure 448).

Figure 448 APPN Routed Hardware Configuration



To configure port 4 on NETBuilder bridge/router A for these settings, follow these steps:

- Set up APPN according to the information in the Configuring APPN Intermediate Session Routing chapter.
- Add a filter policy named "APPNPolicy" with `PROTOCOLRsrv` as the action option, with the name tag "APPNtag" and with the built-in mask "APPN" by entering:

```
ADD -Filter POLicy APPNPolicy PROTOCOLRsrv APPNTAG APPN
```



When you configure filters to control protocol reservation, the filters will affect all ports on the bridge/router. You can configure filters for specific ports, but this configuration is not recommended for use with protocol reservation.

- Apply the filtering policy by entering (LLC2 is used as the filter type for SNA, NetBIOS, and APPN traffic):
- Set the Filter CONTROL parameter to Enable by entering:
- Configure port 4, with the "APPNtag" `PROTOCOLRsrv` name tag entered in the filter policy in step 2, with 75 percent of reserved bandwidth by entering:

```
SETDefault -Filter SElection = LLC2
```

- Set the Filter CONTROL parameter to Enable by entering:

```
SETDefault -Filter CONTROL = Enable
```

- Configure port 4, with the "APPNtag" `PROTOCOLRsrv` name tag entered in the filter policy in step 2, with 75 percent of reserved bandwidth by entering:

```
ADD !4 -PORT PROTOCOLRsrv APPNTAG 75
```



If configuring protocol reservation on a WAN Extender port, enter the `PROTOCOLRsrv` command specifying a virtual port instead of a physical port.

- Set `PROTOCOLRsrv` as the `-PORT QueueCONTROL` parameter option for port 4 by entering:

```
SETDefault !4 -PORT QueueCONTROL = PROTOCOLRsrv
```

Protocol Reservation Configuration Examples

This section provides protocol reservation configuration examples that use the configuration procedures described earlier in this chapter.

Example 1: Mixed Bridged Traffic

In this example, you are configuring protocol reservation on WAN port 1 that supports SNA- and NetBIOS- bridged traffic through several DLSw tunnels and IP-bridged traffic at the same time.

The hardware configuration is as follows (see Figure 449):

- Bridge/Router A, B, C, and D run DLSw.
- Bridge/Router A has DLSw tunnels with bridge/router B, C, and D.
- Bridge/Router A and bridge/router D run IP.

Figure 449 Mixed-Bridged Traffic Hardware Configuration

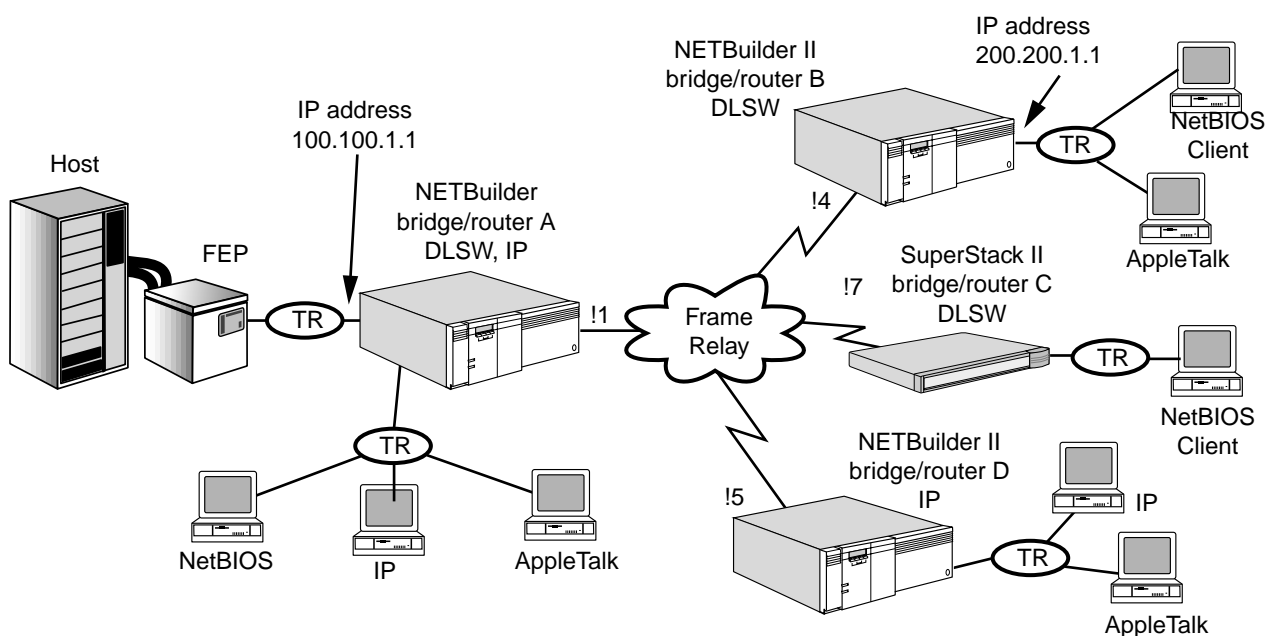


Table 105 lists the port bandwidth for different bridged traffic types for the ports on each bridge/router.

Table 105 Traffic Type and Reserved Bandwidth for Example 1

| Traffic Type | Reserved Bandwidth |
|--|--------------------|
| <u>NETBuilder Bridge/Router A port 1:</u> | |
| Default (AppleTalk traffic) | 5 percent |
| DLSw traffic to DLSwPeer Bridge/Router B with IP address 200.200.1.1 (SNA) | 20 percent |
| All other DLSw traffic (NetBIOS) | 50 percent |
| Traffic to IP hosts | 25 percent |
| <u>NETBuilder Bridge/Router B port 4:</u> | |
| DLSw traffic to DLSwPeer Bridge/Router A with IP address 100.100.1.1 (SNA) | 45 percent |
| DLSw traffic to all other DLSw nodes | 45 percent |
| <u>NETBuilder Bridge/Router C port 7:</u> | |
| DLSw traffic to DLSwPeer Bridge/Router A with IP address 100.100.1.1 (NetBIOS) | 95 percent |
| <u>NETBuilder Bridge/Router D port 5:</u> | |
| IP traffic | 45 percent |
| AppleTalk traffic | 45 percent |

Table 106 lists all the commands required to configure protocol reservation on each of the bridge/routers shown in the figure. Before entering the commands in the table, configure bridging (see the Configuring Bridging chapter), and DLSw (see the Configuring Data Link Switching for SNA and NetBIOS Networks chapter).

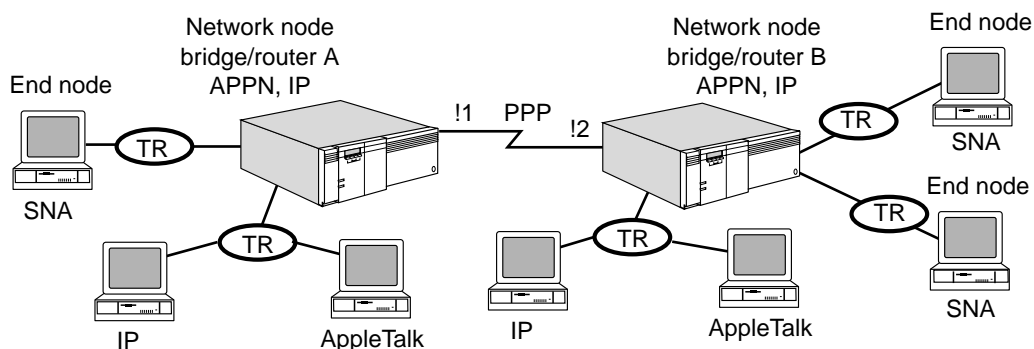
Table 106 Required Commands (Example 1)

| Commands Entered on Bridge/Router A | Commands Entered on Bridge/Routers B, C and D |
|--|---|
| <pre>ADD -Filter POLicy IP POLICY PROTOcolRsrv IPTag IP SETDefault -Filter SElection = BRIdged SETDefault -Filter CONTrol = Enable ADD !1 -PORT PROTOcolRsrv IPTAG 25 ADD !1 -PORT PROTOcolRsrv DLSwPeer 200.200.1.1 20 ADD !1 -PORT PROTOcolRsrv DLSw 50 SETDefault !1 - PORT QueueCONTrol = PROTOcolRsrv</pre> | <pre><u>Bridge/Router B:</u> ADD !4 -PORT PROTOcolRsrv DLSwPeer 100.100.1.1 45 ADD !4 -PORT PROTOcolRsrv DLSw 45 SETDefault !4 - PORT QueueCONTrol = PROTOcolRsrv <u>Bridge/Router C:</u> ADD !7 -PORT PROTOcolRsrv DLSwPeer 100.100.1.1 95 SETDefault !7 - PORT QueueCONTrol = PROTOcolRsrv <u>Bridge/Router D:</u> ADD -Filter POLicy IP Policy1 PROTOcolRsrv IPTAG IP SETDefault -Filter SElection= BRIdged SETDefault -Filter CONTrol = Enable ADD !5 -PORT PROTOcolRsrv IPTAG 45 ADD -Filter POLicy Policy2 PROTOcolRsrv any_Apple ATALK ADD !5 -PORT PROTOcolRsrv any_Apple 45 SETDefault !5 - PORT QueueCONTrol = PROTOcolRsrv</pre> |

**Example 2:
Mixed-Routed Packets**

In this example, you are configuring protocol reservation on a WAN port for APPN-routed and IP-routed traffic at the same time. AppleTalk-routed packets use the 5 percent default bandwidth. Figure 450 shows the hardware configuration for this example.

Figure 450 Mixed Routed Packets Hardware Configuration



The goal in this example is to divide the bandwidth of the port for different types of transmitting traffic from the port as shown in Table 107.

Table 107 lists the port bandwidth for different routed traffic types for the ports on each bridge/router.

Table 107 Traffic Type and Reserved Bandwidth for Example 2

| Traffic Type | Reserved Bandwidth |
|---|--------------------|
| <u>NETBuilder Bridge/Router A port 1:</u> | |
| Default (AppleTalk traffic) | 5 percent |
| APPN-routed traffic | 70 percent |
| IP-routed traffic | 25 percent |
| <u>NETBuilder Bridge/Router B port 2:</u> | |
| Default (AppleTalk traffic) | 5 percent |
| APPN-routed traffic | 70 percent |
| IP-routed traffic | 25 percent |

Table 108 lists all the commands required to configure protocol reservation on each of the bridge/routers shown in the figure. Before entering the commands in the table, configure IP routing (see the Configuring IP Routing chapter), APPN routing (see the Configuring APPN Intermediate Session Routing chapter), and AppleTalk routing (see the Configuring AppleTalk Routing chapter).

Table 108 Required Commands (Example 2)

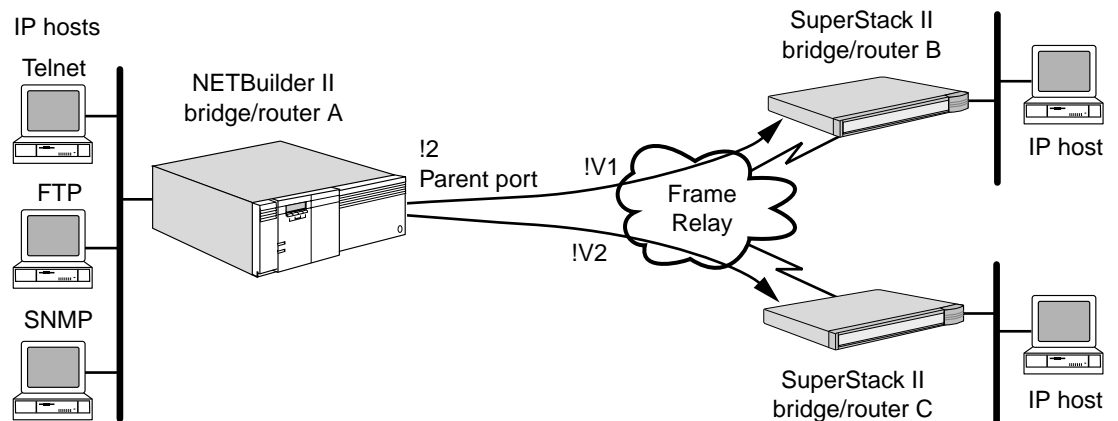
| Commands Entered on Bridge/Router A | Commands Entered on Bridge/Router B |
|---|---|
| ADD -IP FilterAddrs ALL ALL PROTOcolRsrv IPTAG IP | ADD -IP FilterAddrs ALL ALL PROTOcolRsrv IPTAG IP |
| SETDefault -IP FilterDefAction = Forward | SETDefault -IP FilterDefAction = Forward |
| SETDefault -IP CONTROL = Filtering | SETDefault -IP CONTROL = Filtering |
| ADD -Filter POLicy APPNPolicy PROTOcolRsrv APPNTAG APPN | ADD -Filter POLicy APPNPolicy PROTOcolRsrv APPNTag APPN |
| SETDefault -Filter SELECTION= LLC2 | SETDefault -Filter SELECTION= LLC2 |
| ADD !1 -PORT PROTOcolRsrv APPNTAG 70 | ADD !2 -PORT PROTOcolRsrv APPNTag 70 |

Table 108 Required Commands (Example 2)

| Commands Entered on Bridge/Router A | Commands Entered on Bridge/Router B |
|--|---|
| <code>ADD !1 -PORT PROTOcolRsrv IPTAG 25</code> | <code>ADD !2 -PORT PROTOcolRsrv IPTAG 25</code> |
| <code>SETDefault !1 -PORT QueueCONTRol = PROTOcolRsrv</code> | <code>SETDefault !2 - PORT QueueCONTRol = PROTOcolRsrv</code> |

Example 3: Virtual Ports In this example, when you configure protocol reservation for Frame Relay virtual ports, you configure the protocol reservation on the parent port, not on the virtual port. No special configuration for virtual ports is necessary if you do not specify port numbers when you set up the filter masks.

Figure 451 shows a configuration with virtual ports. In this configuration, on NETBuilder bridge/router A, you configure the specific filter masks without specifying a port number. You then configure the filter percentages using the `-PORT PROTOcolRsrv` parameter on port 2, the parent port to virtual ports 1 and 2.

Figure 451 Protocol Reservation on Virtual Ports

How Protocol Reservation Works

Protocol reservation allows you to assign a percentage of bandwidth to designated packets transmitting through a specified WAN logical port (no virtual ports) and meeting certain conditions. The conditions can include protocol type, packet length, and packets destined for a specified address, among others.

Protocol reservation allows you to reserve bandwidth for lower bandwidth usage, interactive, response-time-sensitive, or transaction-oriented network application packets. These type of packets are normally crowded out by heavy bandwidth usage applications such as file transfer or mail.

For example, in a multiprotocol environment that includes IBM protocol traffic (such as response-time-sensitive SNA packets) mixed in with other protocol traffic (such as IP or IPX), SNA devices throttle back the data transmission rate to the end station when they sense available bandwidth decreasing. If other network protocols increase this bandwidth consumption, SNA devices will throttle back the data transmission rate more, which slows the response time of SNA packets even more.

To avoid this situation, use protocol reservation to provide a percentage of bandwidth for the SNA packets and to restrict the percentage of bandwidth to the other more aggressive protocol packets to ensure that the small, response-time-sensitive SNA packets to pass through port in a timely manner.

How Protocol Reservation Controls Bandwidth for Traffic Types

Protocol reservation provides a “valve” that sits above the transmit queue and controls the amount of bandwidth reserved for specific types of data (identified by the user through filtering schemes). During times of stress when bandwidth of the WAN link is consumed beyond a threshold, the protocol reservation valve engages and works to normalize ratios of traffic types down to the configured percentages. If the bandwidth is not utilized to threshold, the NETBuilder bridge/router does not attempt to achieve the configured percentages so all packets can be serviced.

The traffic is identified by the user through the use of mnemonic or manually configured filters. OSI, DECnet, Vines, AppleTalk, and XNS routed traffic is treated as “default”; specific percentages cannot be assigned to them (percentages can be allocated if these protocols are bridged). When the protocol reservation valve is engaged each packet is checked against the configured filter(s), which introduces some latency when compared with traffic flow over an uncongested link. However reserving minimum percentages for a protocol helps to prevent session loss, which may occur during traffic bursts.

Some protocols reduce their transmit rate when congestion is sensed, and thus may not use all of their allocated bandwidth. Protocol reservation automatically allocates any unused bandwidth to other protocols, and the desired effect may not be achieved. If you are using a protocol that reduces its transmit rate and does not utilize its configured bandwidth, you may want to use the priority queueing feature, which allows you to control the order in which packets are serviced between high, medium, and low priority queues (see the Prioritizing Multiprotocol Data chapter).

Protocol reservation supports PPP virtual ports or WAN Extender virtual ports. For Frame Relay and SMDS, protocol reservation is supported only on the parent ports at the physical port level (specific percentages are not applied to individual virtual ports, except for WAN Extender ports).

Tuning

The protocol reservation valve normalizes bandwidth to configured percentages over time intervals. If a large packet is encountered, the packet must be passed to the driver transmit queue in its entirety; it is not fragmented into smaller sizes. Other packets will be passed to the queue behind it. Percentages are maintained over time but it is still possible for some traffic to experience latency in extremely busy environments if larger packets fill up the driver transmit queue. Do not allocate more bandwidth for a protocol than you can use. If a protocol cannot reach the percentage of bandwidth allocated to it, then the bandwidth not used by the protocol will be used by other protocols.

Bandwidth Allocation Process Rules

Protocol reservation uses bandwidth allocation process rules for allocating bandwidth to WAN ports.

Bandwidth Normalization

The protocol reservation features can be used for IP routing, IPX-routing, all bridging protocol traffic, and all NETBuilder-provided IBM traffic. To allow any

traffic that is not "tagged" to go through, at least 5 percent of the bandwidth is reserved for this untagged traffic. This reservation is called the *default queue*. Untagged traffic includes non-bridged AppleTalk, XNS, OSI, DecNet, and VINES protocol packets.

If the total configured bandwidth percentages for the port exceed 95 percent, the values are balanced by the system so that the default queue still has its default allotment of approximately 5 percent of the available bandwidth. The rest of the bandwidth is distributed among the entries configured for the port in a ratio to the percentages that were configured for each.

This process of distributing the ratio is called *normalization*. Since the distribution only uses whole numbers for a percentage, the fraction remainders of each protocol are added to the default queue. As a result, the default queue sometimes can have a percentage greater than 5.

Table 109 is an example of the traffic type, configured bandwidth, and then the normalized numbers that occur when the configured assignments of bandwidth exceed 95 percent.

Table 109 Traffic Type, Configured Bandwidth, and Normalized Bandwidth

| Traffic Type | Configured Bandwidth | Normalized Bandwidth |
|-----------------------------|----------------------|----------------------|
| Default (AppleTalk traffic) | 5 percent | 7 percent |
| SNA traffic | 95 percent | 37 percent |
| IP traffic | 95 percent | 37 percent |
| IPX traffic | 55 percent | 19 percent |

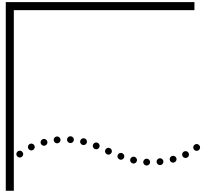
Distribution of Non-Allocated Bandwidth

If the total configured bandwidth percentages are less than 95 percent, the non-allocated bandwidth is added to the default to be given to the configured protocols or for untagged traffic on a first-come first-served basis.

For example, if you configure protocol reservation for a WAN port with the following bandwidth allocations:

- 50 percent of the bandwidth for SNA traffic
- 20 percent of the bandwidth for NetBIOS traffic
- 5 percent automatically set aside as default bandwidth for untagged traffic

The remaining 25 percent of the bandwidth is added to the default to be used for SNA traffic, NetBIOS traffic, or for untagged traffic, whatever traffic needs it first.



CONFIGURING DATA COMPRESSION

Data compression is an optional feature that may be used to enhance the effective throughput on Point-to-Point Protocol (PPP), X.25, and Frame Relay connections.



If you are using a modem that already performs compression, 3Com suggests that you do not configure data compression.

Three data compression types exist: tinygram, history-based, and per-packet. Tinygram compression is a packet-level compression that can be configured for PPP links only. History-based and per-packet compression are link-level compression types. History-based compression may be configured for PPP and X.25. Per-packet based compression may be configured for PPP, X.25, and Frame Relay. All three types of compression operate and are configured independently.

This chapter describes how to configure these compression types and how data compression works and when to use each type.



For conceptual information, see "How Data Compression Works" later in this chapter.

Configuring Data Compression

The following procedures describe how to configure tinygram, history, and per-packet compression.

Configuring Tinygram Compression

The `-PATH TinyGramcomp` parameter allows you to compress all bridged Ethernet packets that are 64 bytes and are padded with trailing zeros. When the packet is sent on a serial line, the receiving side reinserts the zeros before forwarding the packet to an Ethernet LAN. This compression method is effective only on serial lines and is normally used in the Digital Equipment Corporation/local area transport (DEC/LAT) terminal-to-host environments.

This procedure must be completed at both ends of the link. To enable tinygram compression, follow these steps:

- 1 Enable tinygram compression on a specific path using:

```
SETDefault !<path> -PATH TinyGramcomp = Enabled
```

- 2 Verify the PATH configuration using:

```
SHowDefault !<path> -PATH TinyGramcomp
```

- 3 Activate tinygram compression using:

```
SETDefault !<path> -PATH CONTrol = Enabled
```

Configuring Link-Level Compression

Link-level compression can be configured as either history-based or packet-based. History-based link-level compression requires a reliable link for proper operation.

PPP and X.25 support both history and per-packet compression. When history-based compression is enabled over a PPP link, Link Access Procedure, Balanced (LAPB) must be run to provide the reliable link. Frame Relay does not provide a reliable link and therefore does not support history-based compression.

Per-packet compression is supported for PPP, X.25, and Frame Relay links.

Enabling History-based or Per-packet Compression

This procedure must be completed at both ends of the link. To enable history-based or per-packet link-level compression, follow these steps:

- 1 Enable link-level history based compression on a particular port using:

```
SETDefault !<port> -PORT COMPResType = HISTory
```



Frame Relay does not support history-based data compression.

- 2 To select the per-packet link-level compression algorithm, use:

```
SETDefault !<port> -PORT COMPResType = PerPacket
```

- 3 For the configured compression types to take effect, re-enable the port using:

```
SETDefault !<port> -PORT CONTRol = Enabled
```

Before you re-enable the port with this command, you must perform any optional configuration steps such as those described in “Enabling LAPB for a PPP Link,” “Frame Relay Configuration Options,” and “X.25 Configuration Options” later in this chapter.

Optional Configurations The following sections describe optional configurations you may perform.

Enabling LAPB for a PPP Link

If you are planning to run the history-based algorithm on a PPP link, set up LAPB by follow these steps:

- 1 Enable LAPB at both ends of the link using:

```
SETDefault !<path> -LAPB CONTRol = Enable
```

- 2 Configure one end of the serial link as data terminal equipment (DTE) using:

```
SETDefault !<path> -LAPB InterfaceType = DTE
```

- 3 Configure the other end of the serial link as data communications equipment (DCE) using:

```
SETDefault !<path> -LAPB InterfaceType = DCE
```

Frame Relay Configuration Options

When you set up data compression for Frame Relay links, data compression can be specified for each individual data link connection identifier (DLCI). The -FR COMPResType parameter overrides the -PORT COMPResType parameter.

Over Frame Relay connections, data compression is not negotiated. Data compression must be configured appropriately at both ends of the link.

- To enable per-packet data compression on a specific DLCI in an Frame Relay link, use:

```
SETDefault !<port> -FR COMPResType = <dlci> PerPacket
```

- To disable data compression on a specified DLCI, use:

```
SETDefault !<port> -FR COMPResType = <dLci> NONE
```
- To use the compression type configured by the -PORT COMPResType parameter, use:

```
SETDefault !<port> -FR COMPResType = <dLci> DEFault
```

The DEFault value for the -FR COMPResType parameter in this command requires that the DLCI use the compression type configured on the PORT Service.

X.25 Configuration Options

When you set up data compression for X.25 links, data compression can be specified for individual X.25 profiles. The -PROFile X25COMPResType parameter overrides the -PORT COMPResType parameter.

- To enable per-packet data compression on a specific profile on an X.25 link, use:

```
SETDefault !<profile ID> -PROFile X25COMPResType = PerPacket
```
- To enable history data compression on a specific profile on an X.25 link, use:

```
SETDefault !<profile ID> -PROFile X25COMPResType = HIStory
```
- To disable data compression on a specified profile, use:

```
SETDefault !<profile ID> -PROFile X25COMPResType = NONE
```
- To use the compression type configured for the PROFile Service X.25 link, use:

```
SETDefault !<profile ID> -PROFile X25COMPResType = DEFault
```

The DEFault value for the -PROFile X25COMPResType parameter in this command requires that the selected profile use the compression type configured on the PORT Service.

Verifying Link-Level Compression Effectiveness

In multiprotocol and mixed-application environments, it may be difficult to achieve a consistent high level of compression. The effectiveness of compression is measured by the ratio of the uncompressed data to the compressed data, also known as the compression ratio. The software can display the number of raw and compressed bytes, which you can use to measure the effectiveness of link-level compression for your environment.

After you have configured link-level compression, you may need to determine its effectiveness for your network environment. To decide whether link-level compression is beneficial on your network, follow these steps:

- 1 Display the link compression status using:

```
SHow !<port> -PORT LinkCompStat
```

The system will display statistics accumulated since link compression was configured. The following is a sample display:

```
-----Compression Statistics for Port = 4 LCN = 1-----
Owner      = X25
CompType   = PerPacket
TX_Raw     = 742
TX_Comp    = 200
TX_Ratio   = 3.71
RX_Raw     = 452
RX_Comp    = 162
RX_Ratio   = 2.79
TX_Fail    = 0
RX_Err     = 0
```

If you want to recalculate link compression performance, old statistics can be removed using:

```
FLush !<port> -PORT LinkCompStat
```

- 2 Compare the number of raw bytes to compressed bytes for both the transmit and receive sides; check all ports that are currently using compression.

Compression is very CPU-intensive. You may want to disable compression if you do not get a favorable compression ratio, which depends on the nature of the data, or if the overall system performance suffers because of CPU overloading.

How Data Compression Works

Data compression performs additional processing on the contents of each packet to look for repetitive patterns. Consequently, it is most effective when there is sufficient CPU cycles available to handle the additional processing. Data compression is most effective on slow lines.



If you are using a modem that already performs compression, 3Com suggests that you do not configure data compression.

Tinygram Compression

Tinygram compression is packet-level compression. Tinygram compression is performed on packets with a length of less than 64 bytes. An increase in effective throughput is achieved by suppressing the transmission of trailing nulls, or hexadecimal zeros, in packets that are encapsulated in the Ethernet frame format. This type of compression is called tinygram compression and is also referred to as local area transport (LAT) compression (since LAT packets are typically small in size and are padded up to 64 bytes with trailing nulls). The receiving end of a compressed packet can easily recreate the original packet by adding the trailing nulls. Because the CPU cycles involved in the stripping and adding of the trailing nulls are significantly less than the time it takes to transfer those nulls across a slow-speed line, the effective throughput of the system is increased.

Link-Level Compression

Link-level compression is performed over all packets sent on a specified link. The effective throughput is increased by sending fewer bytes across the link, as with tinygram compression. The algorithm used for link-level compression looks for repetitive data patterns in packets and replaces them with shorter length codes.

The software supports the following types of link-level compression algorithms:

- History-based
- Per-packet

The algorithm used for history-based link-level compression looks for repetitive data patterns across multiple packets and replaces them with shorter length codes. The sending and receiving ends both build up a history buffer, and encode and decode the data in the packet according to that buffer. The history buffer will have the last 2 KB of data. For proper encoding and decoding, the history buffer at each end must always be synchronized. Because the history information is transferred along with compressed data, the sending side must be assured that the receiving side reliably gets the data. As a result, history-based compression can operate *only* over a reliable data link. History-based compression requires that the LAPB Service be configured and operational over all the links on which this type of compression is desired. By default, the history-based link-level compression is selected.

The algorithm used for per-packet link-level compression looks for repetitive patterns within a packet and replaces them with shorter length codes. With per-packet compression, the sending and receiving ends do not preserve the history between packets. As a result, per-packet compression does not need to operate over a reliable data link, and the LAPB Service does not need to be configured over all links on which this type of compression is desired.

Because history-based compression looks for repetitive data across multiple packets, it is more effective in shrinking a packet size, which includes the line throughput. When considering history-based compression, the memory required to maintain a history buffer (approximately 26 KB of memory per interface) must be considered, particularly if it is enabled on several links. Because a history buffer is not maintained in per-packet mode, the memory requirement is considerably less (24 KB of memory per a fully populated NETBuilder II system) than for history mode.

When To Use Tinygram Compression

The decision about whether to use tinygram compression depends on the characteristics of your system. Some general recommendations, which are based on line speeds of 64 kbps, are provided here.

Consider using tinygram compression in the following situation:

- In environments with small data packets using null character padding (for example, in LAT environments)

Avoid using tinygram compression in the following situations:

- In environments where packets are generally transmitted with enough data to create 64-byte packets requiring no padding
- In environments where small data packets use random data for padding (for example, in some Telnet environments)

When To Use Link-Level Compression

The decision about whether to use link-level compression depends on the characteristics of your system. Some general recommendations, which are based on line speeds of 64 kbps, are provided here.

Consider using link-level compression in the following situations:

- In environments with repetitive patterns in the bit stream being transferred (for example, with file transfers or electronic mail)
- In environments where slow lines (64 kbps or lower) are being used

Avoid using link-level compression in the following situations:

- In environments with patterns in the bit stream that are *not* repetitive (for example, in image files)
- In environments where high-speed lines are being used
- When the overall throughput of a system is already below normal

If you decide to use link-level compression, you must further decide which type to use: history-based or per-packet. Some general recommendations, which are based on line speeds of 64 kbps, are provided here.

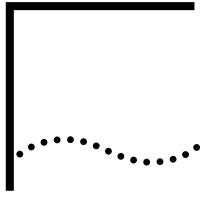
Consider using history-based link-level compression in the following situations:

- In remote sites with 1–2 LAN ports and 1–2 WAN ports or in central sites with minimal LAN traffic and 3–6 WAN ports
- In sites where the wide area links are considered reliable

Consider using per-packet link-level compression in the following situations:

- In central sites with more than 6 WAN ports
- In any site where there is significant LAN-to-LAN traffic

Use the preceding recommendations as general guidelines. In cases where link compression is used, verify the effectiveness of the algorithm with the `-PORT LinkCompStat` parameter. If you do not see a significant difference between the raw and compressed bytes, the serial line throughput increase may not be enough to offset the overhead of applying the algorithm. If overall performance degradation is experienced, you should reevaluate the continued use of link compression.



PRIORITIZING MULTIPROTOCOL DATA

This chapter describes how to use the data prioritization feature to assign a priority (high, medium, or low) to most packets that are forwarded over a wide area network using Point-to-Point Protocol (PPP), Frame Relay, or Switched Multimegabit Data Service (SMDS).



For conceptual information, see “How Data Prioritization Works” later in this chapter.

Advantages of Prioritizing Data

You can receive the following benefits by using the data prioritization feature:

- Control data traffic on heavily used wide area networks.
- Allow the following types of packets to have a higher priority over other data traffic on a wide area network:
 - Network-critical traffic, for example, bridge spanning tree packets (system-configured)
 - Mission-critical traffic, for example, Logical Link Control, type 2 (LLC2) tunnel packets (user-configured)
 - Time-critical traffic, for example, Telnet packets (system-configured)
 - Specific protocol packets, for example, Advanced Peer-to-Peer Networking (APPN) or Internet Protocol (IP)-routed packets (user-configured)
- Allow sessions to be prioritized according to the session characteristics.
- Avoid LLC, Systems Network Architecture (SNA), and NetBIOS session failures due to timeouts.

Setting Up Data Prioritization

This section describes how to set up the data prioritization feature.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Determine what types of packets you want to prioritize and what priority you want to assign to each type.

For example, you may want to assign IP-routed packets a high priority, Internetwork Packet Exchange (IPX) packets a medium priority, and AppleTalk packets a low priority. You also may want to prioritize types of packets within the IP protocol itself.
- Read through this chapter to familiarize yourself with the different ways you can assign a priority to a type of packet. Determine which option you want to use for each type of packet you want to prioritize.

For example, you may want to assign a high priority to IP-routed packets, a medium priority to IPX packets by setting up a mask and policy, and a low priority to all other packets, including AppleTalk packets.

- Determine the interleave factor, which is defined as the ratio of packets that you want forwarded from high- to medium-priority queues and the ratio of packets you want forwarded from medium- to low-priority queues. For more information on the interleave factor and queue arbitration, see “How Data Prioritization Works” later in this chapter.

Procedure To set up the data prioritization feature, follow these steps. The example of assigning a high priority to IP-routed packets, a medium priority to IPX packets, and a low priority to all other packets including AppleTalk packets will be used throughout this procedure.

- 1 If you want to assign a priority to APPN, LLC2 tunnel, or IP-routed packets, use one of the following lines of syntax:

```
SETDefault -APPN QueuePriority = <H | M | L | DEFault>
SETDefault -LLC2 TUNnelPRiority = <H | M | L | DEFault>
SETDefault -IP QueuePriority = <H | M | L | DEFault>
```

For example, to assign a high priority to IP-routed packets, enter:

```
SETDefault -IP QueuePriority = H
```

If you retain the default setting of DEFault for any of the above commands, the system uses the setting of the -PORT DefaultPriority parameter. For instructions on configuring the -PORT DefaultPriority parameter, see step 3.

- 2 If you want to assign a priority to packets other than LLC2 tunnel or IP-routed packets, follow these steps:
 - a Set up a mask that determines the types of packets that should be prioritized.

You can set up either a built-in or a user-defined mask. Since using a built-in mask requires less configuration, 3Com recommends this option. To determine if a built-in mask exists for the type of packet you want to prioritize, see the Configuring Mnemonic Filtering chapter. If a built-in mask that suits your purposes exists, go on to step 2b.

In the example used throughout this procedure, you want to assign a medium priority to IPX packets. Since a built-in mask for IPX packets exists, you do not need to create a mask.

If a built-in mask that suits your needs does not exist, configure one using:

```
ADD -FIlter MASK <maskname> <location> [<operation>] <pattern>
```

For example, to configure a mask that prioritizes packets with a value greater than %45 at the first byte of data, enter:

```
ADD -FIlter MASK some_data dl.data+%0>%45
```

For more examples of configuring various masks, see the Configuring Mnemonic Filtering chapter.

- b Set up a policy that determines the priority each type of packet should be assigned using:

```
ADD -FIlter POLIcy <policyname> <action> <masks> [<context>]
```

For example, to prioritize IPX packets at a medium priority using a built-in mask, enter:

```
ADD -FIlter POLIcy prioritize_ipx PRIoritize M IPX
```

- 3 The default priority for packets that you do not specifically assign a priority to (as in steps 1 and 2) is medium. If you want to change this priority, use:

```
SETDefault -PORT DefaultPriority = <H | M | L>
```

For example, to change the default priority for all packets other than IP and IPX, including AppleTalk packets, enter:

```
SETDefault -PORT DefaultPriority = L
```

- 4 The default ratio of packets forwarded from high- to medium-priority queues is 3 and the ratio of packets forwarded from medium- to low-priority queues is 2. If you want to reconfigure these ratios, use:

```
SETDefault !<port> -PORT QueueInterLeave = <ratio1> <ratio2> (1-10)
```

For example, to change the default values to 6 and 3 on port 3, enter:

```
SETDefault !3 -PORT QueueInterLeave = 6 3
```

For more information on forwarding ratios and queue arbitration, see “How Data Prioritization Works” later in this chapter.

- 5 By default, the serial line driver will accept as many medium- and low-priority packets as it can possibly handle. If you anticipate that high-priority packets, such as SNA packets, may be slowed down by many medium- and low-priority packets, (especially large medium- and low-priority packets) in the queue, adjust the number of medium- and low-priority packets forwarded to the queue using:

```
SETDefault !<port> -PORT QueueThrottle = <1-40>
```

3Com recommends setting the value between 1 and 20 if you are using a slow-speed line (64K or below) and between 21 and 40 if you are using a high-speed line (for example, T1).

For more information on the parameters used in this procedure, see the Filter Service Parameters chapter, the IP Service Parameters chapter, the LLC2 Service Parameters chapter, and the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

Prioritizing LLC2-, SNA-, and NetBIOS-Bridged Packets

This section provides information on assigning a priority to the following types of packets using the data prioritization feature:

- LLC2-bridged packets from two groups of end stations (one group simulates 3270 interactive traffic; the other, SNA file transfers)

The setting of the -LLC2 TUNnelPRiority parameter on bridge/routers located at both ends of the tunnel should be the same. For example, if the setting of the -LLC2 TUNnelPRiority parameter on bridge/router 1 located on one end of the tunnel is high, then the setting of this parameter on bridge/router 2 located on the other end of the tunnel should also be high.

- SNA-bridged packets
- NetBIOS-bridged packets

To assign a priority to these types of packets, you need to set up a filter, which includes a mask and a filter policy. The mask specifies the type of packet that should be prioritized; the filter policy determines the priority that the specified packet should be assigned.

The following sections provide more information on assigning priorities to these types of packets.



CAUTION: *Do not prioritize connection-oriented packets such as LLC2 (SNA, NetBIOS, etc.) to the low queue because the low queue can be flushed to favor high and medium packets. With connection-oriented packets such as LLC2, REjects and possible session disconnects will be generated.*

Prioritizing LLC2-Bridged Packets From Two Groups of End Stations

Suppose you want to assign LLC2-bridged packets from end station group 1 (3270 interactive traffic) a high priority and LLC2-bridged packets from end station group 2 (SNA file transfers) a medium priority. You also want to assign all other packets a low priority.

Since you want to prioritize a certain type of packet received from two groups of end stations, you need to define each group by identifying each end station that belongs to a group using the `-Filter StationGroup` parameter. For example, if groups 1 and 2 are composed of three end stations each, enter:

```
ADD -Filter StationGroup group_1 %0800020000a1
ADD -Filter StationGroup group_1 %0800020000a2
ADD -Filter StationGroup group_1 %0800020000a3
ADD -Filter StationGroup group_2 %0800020000b1
ADD -Filter StationGroup group_2 %0800020000b2
ADD -Filter StationGroup group_2 %0800020000b3
```

Next you need to set up a mask and a filter policy for the LLC2-bridged packets received from end systems in groups 1 and 2. To set up a mask called "inter" that looks for bridged LLC2 packets from end systems in group 1 and a mask called "ft" that looks for bridged LLC2 packets from end systems in group 2, enter:

```
ADD -Filter MASK inter dl.sa = group_1
ADD -Filter MASK ft dl.sa = group_2
```

To set up a filter policy called "interhigh" that assigns a high priority to packets specified in the mask called "inter" and a filter policy called "ftlow" that assigns a low priority to packets specified in the mask called "ft," enter:

```
ADD -Filter POLicy interhigh PRIoritize H inter
ADD -Filter POLicy ftmed PRIoritize M ft
SETDefault -PORT DefaultPriority = L
```

Because you changed the port default priority to low, all other packets are assigned a low priority.

Prioritizing SNA- and NetBIOS-Bridged Packets

Suppose you want to assign SNA-bridged packets a high priority and NetBIOS-bridged packets a medium priority. You also want to assign all other packets a low priority.

You need to set up a mask and a filter policy for the SNA- and NetBIOS-bridged packets. To set up a mask called "sna" that looks for bridged SNA packets and a mask called "netbios" that looks for bridged NetBIOS packets, enter:

```
ADD -Filter MASK sna dl.lsap = %4
ADD -Filter MASK netbios dl.lsap = %f0
```

To set up a filter policy called "snahigh" that assigns a high priority to packets specified in the mask called "sna" and a filter policy called "nbmed" that assigns a medium priority to packets specified in the mask called "netbios," enter:

```
ADD -Filter POLicy snahigh PRIoritize H sna
```

```
ADD -Filter POLicy nbmed PRIoritize M netbios
SETDefault -PORT DefaultPriority = L
```

Because you changed the port default priority to low, all other packets are assigned a low priority.

Assigning a Priority to Different IP Packets

You can use the IP filter facilities in the IP Service to prioritize IP traffic over the traffic of other protocols or to prioritize various types of IP traffic. By using the `-IP FilterAddr`s parameter, you can specify a packet filtering policy. Use the `Qpriority`, `X25Profile`, and `DodDiscard` actions of the `FilterAddr`s parameter for data prioritization among IP packets.

For example, suppose you want to assign a high priority to Telnet packets going to host 129.0.0.1. The socket number used by the Telnet protocol is 23. Enter:

```
ADD -IP FilterAddr ALL> 129.0.0.1 QPriority High 23
```

For more information about the `FilterAddr`s parameter, see the IP Service Parameters chapter in *Reference for Enterprise OS Software*. For more examples using the `Qpriority`, `X25Profile`, and `DodDiscard` actions of the `FilterAddr`s parameter, see the Configuring IP Routing chapter.

Data Prioritization Parameters

Table 451-1 briefly summarizes the available prioritization parameters and the services where they exist.

Table 451-1 Data Prioritization Parameters

| Service | Parameter | Description |
|--------------------|-----------------|---|
| Filter | MASK | Sets up a mask that determines which packets should be prioritized. |
| | POLicy | Sets up a policy that determines the priority a particular type of packet is assigned. |
| APPN
IP
PORT | QueuePriority | The <code>-APPN</code> and <code>-IP QueuePriority</code> parameters set the priority of APPN and IP-routed packets, respectively; if this parameter is set to <code>DEFault</code> , the system uses the setting of the <code>-PORT DefaultPriority</code> parameter.

The <code>-PORT QueuePriority</code> parameter displays the settings of the following parameters: <ul style="list-style-type: none"> ■ <code>-APPN QueuePriority</code> ■ <code>-IP QueuePriority</code> ■ <code>-LLC2 TUNnelPRiority</code> ■ <code>-PORT DefaultPriority</code> |
| IP | FilterAddr | The <code>QPriority</code> option for the <code>FilterAddr</code> s parameter specifies a queue priority value of high, medium, or low. A numerical value specifies an X25 profile ID to be used. For more information on prioritizing packets over the X25 Service, see the <code>PROFile Service Parameters</code> chapter in <i>Reference for Enterprise OS Software</i> and to the <code>Configuring Wide Area Networking Using X.25</code> chapter in this guide. |
| LLC2 | TUNnelPRiority | Sets the priority of LLC2 packets tunneled over an IP internetwork; if this parameter is set to <code>DEFault</code> , the system uses the setting of the <code>-PORT DefaultPriority</code> parameter.

The priority of LLC2 tunnel packets is maintained across 3Com bridge/routers that the packets traverse through the use of the type of service (TOS) field in the IP header. |
| PORT | DefaultPriority | Sets the default priority of packets if one of the following conditions apply: <ul style="list-style-type: none"> ■ The <code>-APPN QueuePriority</code> parameter, the <code>-IP QueuePriority</code> parameter, or the <code>-LLC2 TUNnelPRiority</code> parameter is set to <code>DEFault</code>. ■ A mask and prioritization policy is not configured for a particular type of packet. |

Table 451-1 Data Prioritization Parameters (continued)

| Service | Parameter | Description |
|---------|---|---|
| | QueueInterLeave | Sets and displays the interleave factor, which is defined as the forwarding ratio of high- to medium-priority packets and of medium- to low-priority packets. |
| | QueuePATtern | Displays the interleave factor configured by the -PORT QueueInterLeave parameter translated by the system into a high, medium, and low pattern. A ratio based on the high, medium, and low pattern also displays. |
| | QueueThrottle | Controls the number of medium- and low-priority packets that are forwarded to the driver each time packets from the priority queue are forwarded on to the wide area network. |
| PROFile | X25PacketSiZE
X25PROFileType
X25VCLimit
X25VCQueueSize
X25VCThruputCl
ass
X25VCTimer
X25WindowSiZe | These parameters help you prioritize traffic if you are using the X25 Service. For more complete information on these parameters, see the Configuring Wide Area Networking Using X.25 chapter in this guide and the PROFile Service Parameters chapter in <i>Reference for Enterprise OS Software</i> . |

For more information on these parameters, see the Filter Service Parameters chapter, the IP Service Parameters chapter, the LLC2 Service Parameters chapter, and the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.

How Data Prioritization Works

The bridge/router software implements a prioritization scheme that assigns a priority to each packet then forwards the packet to an urgent-, high-, medium-, or low- priority queue. The packets are then forwarded from the queues onto a wide area network using PPP, Frame Relay, or Switched Multimegabit Data Service (SMDS) in an order controlled by a queue arbitration algorithm.

The system assigns a priority to some packets. Table 110 lists these packet types and the priority assigned to them by the system. You cannot reconfigure the priority of these packets.

Table 110 Packets with System-Assigned Priorities

| Packet Type | Priority Level |
|-----------------------|----------------|
| Bridge Spanning Tree | Urgent |
| DECnet routing update | High |
| OSPF | High |
| Telnet | High |

The system assigns a priority to Telnet packets that originate from the box; it does not assign a priority to routed Telnet packets.

You can assign a priority to all other types of packets that are not listed in Table 110. You can assign priorities to different types of packets in the following ways:

- For APPN, LLC2 tunnel and IP-routed packets, you can assign a high, medium, or low priority using the SETDefault -APPN QueuePriority, SETDefault -LLC2 TUNnelPriority, or SETDefault -IP QueuePriority commands, respectively.
- For all packets other than LLC2 tunnel and IP-routed packets, for example, AppleTalk packets, you can set up a mask that determines what packets should be prioritized and a policy that determines what priority the packets should be assigned.

Any packet not specifically assigned a priority receives its priority from the setting of the `-PORT DefaultPriority` parameter. The default setting of this parameter is medium.

When multiple ports are attached to one path, no one particular port receives a higher priority over another port. All ports attached to one path receive the same priority.

For complete information on assigning priorities to packets, see “Setting Up Data Prioritization” earlier in this chapter. For more information on the parameters described in this section, see *Reference for Enterprise OS Software*.

How Packets Are Assigned a Priority

All packets are assigned either an urgent, high, medium, or low priority before they are transmitted over a wide area interface.

The system assigns a priority to a packet using the following process:

- 1 The system assigns a priority to certain packet types (see Table 110). You cannot change the priority assigned to these packets.
- 2 The system assigns a priority to APPN, IP-routed, and LLC2 tunnel packets with the settings of the `-APPN QueuePriority`, `-IP QueuePriority`, and `-LLC2 TUNnelPRiority` parameters. You can change the default settings of these parameters.
- 3 The system assigns a priority to all packets other than APPN, IP-routed, and LLC2 tunnel packets through a prioritization filter.

You set up a prioritization filter by configuring masks that determine the types of packets that should be prioritized using the `ADD -Filter MASK` command. A configuration policy that determines the priority each type of packet should be assigned using the `ADD -Filter POLicy` command.

- 4 The system assigns a priority to all other packets that do not receive their priority through any of the previously discussed methods via the setting of the `-PORT DefaultPriority` parameter. You can change the default setting of this parameter.

After the system assigns a priority to a packet using one of steps described above, the packet is forwarded to a queue for transmission to a wide area network. For example, if a packet is assigned a priority using the method described in step 2, the packet does not undergo the methods described in steps 3 and 4. The system handles packet priority assignment in this way to reduce the number of packets that are sent through the prioritization filter (step 3) because filtering can impact system performance.

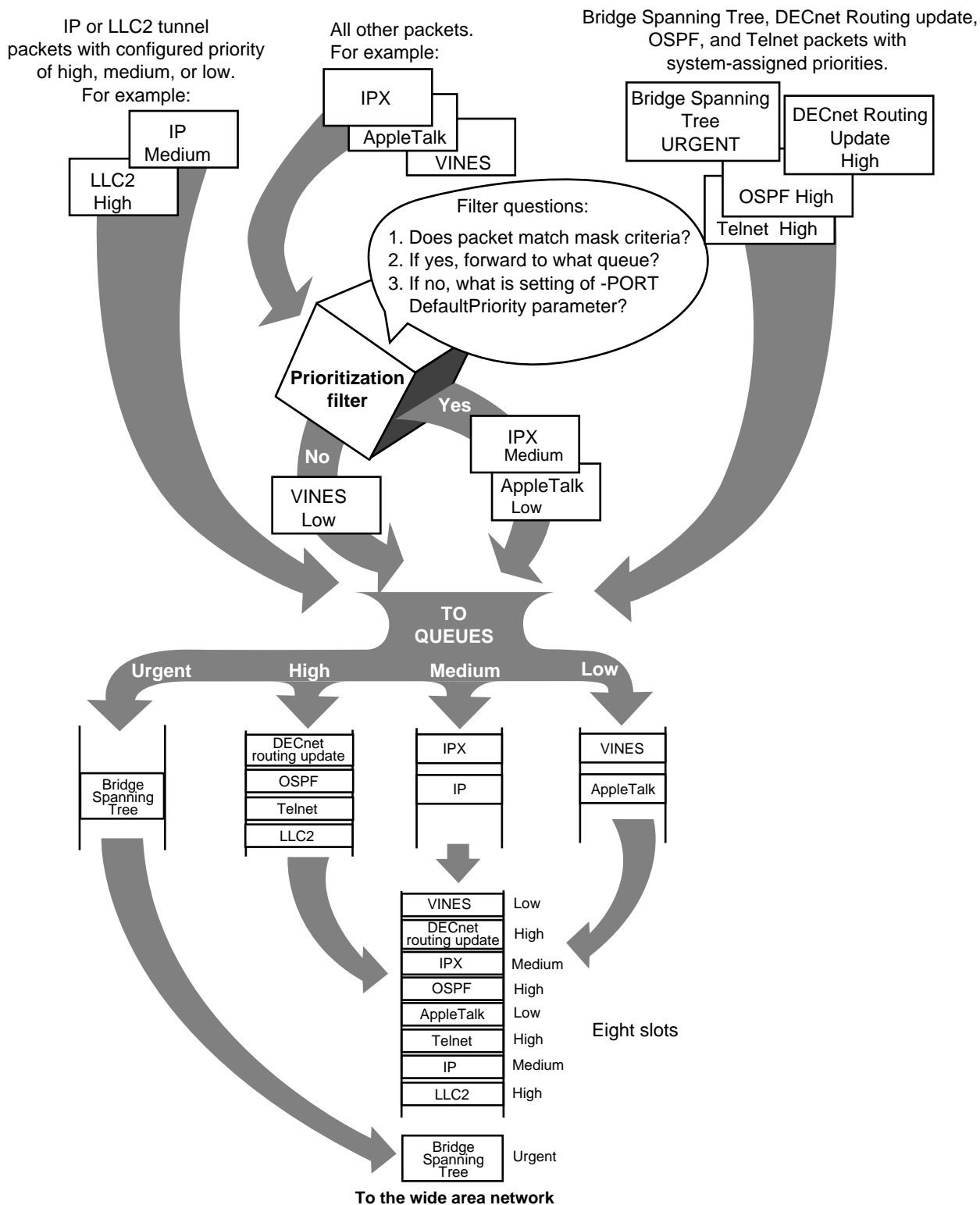
Queues

After the system assigns a priority to a packet automatically or through configured or default parameter settings, the packet is forwarded to one of four types of queues: urgent, high, medium, or low. Figure 452 shows how particular types of packets are assigned priorities and how they are forwarded to the various queues. This figure assumes that a user has configured:

- A medium priority for IP-routed packets.
- A high priority for LLC2 tunnel packets.
- A filter that assigns a medium priority to IPX packets.
- A filter that assigns a low priority to AppleTalk packets.
- The value of low for the `-PORT DefaultPriority` parameter.

The other packets are assigned their priorities by the system.

Figure 452 Prioritizing and Forwarding Packets to Queues



The system forwards all packets in the urgent-priority queue before packets in the high-, medium-, and low-priority queues. You cannot assign the urgent priority to a packet. The system automatically assigns an urgent priority to bridging spanning tree packets only.

After all packets in the urgent-priority queue are forwarded, the system forwards packets from the high-, medium-, and low-priority queues according to a queue arbitration algorithm. For more information on this algorithm, see the next section. The system automatically assigns a high priority to DECnet routing update, open shortest path first (OSPF), and Telnet packets and forwards these types of packets to the high-priority queue.

Queue Arbitration Algorithm

Instead of forwarding all packets from high-priority queues, then all packets from medium-priority queues, and so on, the system uses a queue arbitration algorithm, which ensures that the high-, medium-, and low-priority queues are serviced according to an interleave factor.

You can configure the interleave factor by entering:

```
SETDefault -PORT QueueInterLeave
```

This command allows you to set up the forwarding ratio of high- to medium-priority packets and of medium- to low-priority packets.

The algorithm implements an 8-slot queue that is composed of a variable number of high-, medium-, and low-priority slots as shown in Figure 452. Urgent packets bypass this queue and are immediately forwarded.

The number of high-, medium-, and low-priority slots in the queue are based on the setting of the -PORT QueueInterLeave parameter. The system chooses the closest of five possible 8-slot patterns. Table 111 lists sample values of the -PORT QueueInterLeave parameter and the corresponding 8-slot pattern selected by the system.

Table 111 -PORT QueueInterLeave Parameter Values and 8-Slot Patterns

| Value of -PORT QueueInterLeave | Corresponding 8-Slot Pattern |
|--------------------------------|------------------------------|
| 6, 1 | HHMHHLHH |
| 3, 2* | HHMHMHLH |
| 2, 2 | HMHMHLHM |
| 2, 1 | HMHLHMHL |
| 1, 1 | HMHLHMLM |

* Default value.

The actual transmission rate of high, medium, and low packets on a WAN link may not exactly match the 8-slot pattern because of the packet receive rate and receive pattern. For example, only medium and low packets may arrive in one window of time, and if no high packets are available to send at this time, then the medium and low packets are sent out on the serial line. The low and medium transmission queue can get flushed to favor high-priority packets. For example, in a situation where the packet receive rate is much higher than the WAN link can handle, many packets will be dropped, and with the default queue arbitration pattern of 5 high, 2 medium, and 1 low, the actual transmission ratio may be 6 high, 1 medium, and 1 low.

To display the 8-slot pattern selected by the system, enter:

```
SHow -PORT QueuePATtern
```

A ratio based on the pattern also is displayed. For example, for the default value of the -PORT QueueInterLeave parameter (3, 2), the following display appears:

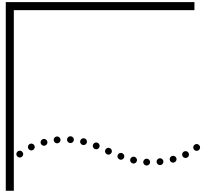
```
HHMHMLH (5:2:1)
```

The contents of this display are based on the setting of the -PORT QueueInterLeave parameter and are the results of a translation algorithm. The contents of this display are calculated by the system and cannot be reconfigured.

This display indicates that the first and second packets are sent from the high-priority queue, the third packet is sent from the medium-priority queue, and so on. Once the eighth packet is sent, the algorithm wraps to the beginning of the pattern again and the first and second packets are sent from the high-priority queue, and so on.

If a packet is sent from the high-priority queue but that particular queue is empty, a packet from the medium-priority queue is sent instead; if the medium-priority queue is also empty, a packet from the low-priority queue is sent instead. If a packet is sent from the medium-priority queue but that particular queue is empty, a packet from the high-priority queue is sent instead. If a packet is sent from the low-priority queue but that particular queue is empty, a high-priority packet is sent instead.

More information on the commands discussed in this section is provided in the PORT Service Parameters chapter in *Reference for Enterprise OS Software*.



NETWORK MANAGEMENT

The bridge/router participates in different types of network management activities. Most management activities require configuration because they are disabled by default. The system manages networks in the following ways:

- Using file service
- Building network maps (netmaps)
- Sending AuditLog messages to a Network Management Station

This chapter describes these management activities, but it does not describe the protocols involved. For information on the protocols, see the appropriate RFCs. For information on the parameters referenced in this chapter, see *Reference for Enterprise OS Software*. The network management information in this chapter applies to the system regardless of its functionality unless otherwise specified.

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) allows you to modify and display some of the bridge/router system parameters from a host; you do not need to attach a terminal to the system console port to change its configuration. The system implementation of SNMP follows the specifications in RFCs 1155, 1157, and 1213. The system parameters and the 3Com-extended parameters that are described in several RFCs including RFC 1213 can both be accessed from the host.

On a 3Com bridge/router, SNMP support is enabled by default. If you want to disable SNMP support, enter:

```
SETDefault -SNMP CONTROL = NoManage
```

Proceed to the next section for instructions on configuring other parameters. All the parameters referenced in this section are SNMP parameters.

Configuring the SNMP Service

This section describes how to configure the SNMP Service.



By default, the community name ANYCOM exists with read access to management information base (MIB) variables and allows unrestricted access to the bridge/router. To ensure that access is available only to the proper system administrator, 3Com recommends that you delete the ANYCOM community name, and add the appropriate community string and the manager's IP address.

Procedure

To configure the bridge/router for SNMP management, follow these steps:

- 1 Delete the default community string "ANYCOM."

For example:

```
DELEte -SNMP COMMunity "ANYCOM"
```

- 2 Configure at least one new community string with read/write access.

For example:

```
ADD -SNMP COMMunity "private" Triv RW ALl
```

- 3 Add other community strings with read-only access as required.

For example:

```
ADD -SNMP COMMunity "public" Triv RO ALl
```

You can have up to ten managers for each community. Including ANYCOM, you can have up to six communities.



When you enter ANYCOM to the list of community names, it must be entered in all uppercase letters.

- 4 Configure at least one SNMP manager to the read/write community string.

For example:

```
ADD -SNMP MANager "private" <IP addr>
```

- 5 SNMP is enabled by default. If SNMP is disabled on your system and you want to enable it, enter:

```
SETDefault -SNMP CONTrol = Manage
```

Related Information

The information in the following sections provide you with additional information about SNMP.

Request Validation

The following options are available with a request for validation:

- For security purposes, the SNMP agent on the system validates SNMP requests before responding. This prevents unauthorized users from viewing or changing the bridge/router configuration.
- You can specify that only the hosts with known community names can send requests. All the community names known to the system are specified by the COMmunity parameter. A request cannot be authenticated if its community name is not included in the COMmunity parameter. To allow requests from any community, add ANYCOM to the list of community names.
- The information in the ANYCOM entry then processes requests with unmatched community names. When adding a community to the list, you can also specify the level of access to the MIB, read or read/write, and the type of trap sent to managers associated with the community name.
- In addition to specifying a set of community names, you can create a list of managers for each community name. If there is no manager list associated with a community name, the system responds to any request with that community name; otherwise, before an incoming request is processed, it must have a matching Internet address for the community name that is specified by the MANager parameter.

Using Traps Traps are sent by the SNMP agent to alert the network management station of unusual events. The following six events are defined by the IETF in RFC1213:

- Cold Start
- Warm Start
- Link Up
- Link Down
- Authentication Failure
- EGP NeighborLoss

With the exception of the Warm Start event, traps are sent each time an event occurs. In addition to these traps, the SNMP agent also sends traps when a Frame Relay virtual circuit changes state, when a bridge port changes spanning tree states, when the station becomes the root of a bridge spanning tree, when an RMON event is triggered, and when an unauthorized Dynamic Host Configuration Protocol (DHCP) or Bootstrap Protocol (BOOTP) server is detected.

Set Operation You can use the SNMP SetRequest protocol data unit (PDU) to set the objects on a bridge/router. The objects may be single or a table entry field (that is, you may specify multiple objects, single or a table entry field, for one PDU). If the request does not follow these restrictions, the SNMP agent responds with a general error PDU, and the set operation is disallowed.

Remote Network Monitoring Alarms

To assist network managers in identifying abnormal activity that could adversely affect network performance, the system contains a Remote Monitoring (RMON) agent based on RFC 1271. The 3Com implementation supports two RMON object groups as follows:

- The Alarm group

You can set up alarms to monitor the MIB objects of interest through the Alarm group; this group includes an Alarm Table, which you must configure before using alarms. 3Com allows five alarms to be configured in the Alarm Table.

- The Event group

You can set up events to either record the monitoring information or notify the network management station. The Event group includes an Event Table and a Log Table. You must configure the Event Table before using it. The Log Table needs no prior configuration; it is a read-only data table for the network management station. 3Com allows five events to be configured in the Event Table.

Because the system contains no user interface to access the RMON agent, you must access it through an SNMP network management station by using SNMP SET/GET/GET-NEXT requests, or by using an SNMP application that can generate SNMP requests. In order to monitor MIB objects of interest according to a set or preset alarms and events, you need to configure the RMON alarms and events with the desired control parameters. The control parameters to be configured include alarm threshold values, which are used to determine if an event needs to be generated. An event is generated if the newly read MIB value has crossed the threshold. The event can take any of the following actions:

- The system sends an SNMP trap to the network management station.

The management station is notified immediately. The management station determines how to react to the SNMP trap.

- The system logs the event into a Log Table in the agent system.
The management station can retrieve the information stored in the Log Table for further analysis. For example, the information collected can be used in selecting proper threshold values.
- The system sends an SNMP trap and logs into the Log Table.

Network Maps

A network map (netmap) contains the Ethernet and Internet addresses of each 3Com device on the attached network and the software version on the device. The bridge/router can participate in building the netmap by broadcasting its addresses to the network at regular intervals (defined by NetMapTime in the SYS Service).

By default, the value of the NetMapTime parameter is set to 0, which means that the system does not broadcast its addresses to the network. To configure the system to broadcast its addresses to the network a specific number of seconds, use:

```
SETDefault -SYS NetMapTime = <number>(0 to 120 seconds)
```

The network map can be used as a network management tool, because you can see at a glance which 3Com devices are on the attached network. To display the network map, enter:

```
SHow -SYS NetMAP
```

For the possible variations for this command and what the various commands will display, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*. For example, to show all the devices that support the TCP/IP Protocol on the network attached to port 1 of the system, enter:

```
SHow !1 -SYS NetMAP tcp
```

The following display appears:

```
NETWORK 1 MAP
0-%08000200F84A    129.213.16.206    1-%08000201136A    129.213.16.208
2-%08000200D26C    129.213.17.152    3-%080002013E86    129.213.18.47
4-%080002013E98    129.213.19.67     5-%080002013E9C    129.213.18.52
6-%0800020128B8    129.213.17.147    7-%08000201ECD3    129.213.16.215
```

To show all the devices that support the Xerox Network Systems (XNS) Protocol on the networks attached to each port of the system, enter:

```
SHow -SYS NetMAP xns
```

The following display appears:

```
3-%080002A01339    NETWORK &00000002 MAP
0-%080002A0133A    NETWORK &00000000 MAP
0-%080002A03C1B    NETWORK &00000003 MAP
0-%080002A01339    NETWORK &00000000 MAP
```


If you want to display the software version running on each 3Com device, use the Long value in the command. For example, enter:

```
SHow !1 -SYS NetMAP Long
```

Because this command does not specify the XNS or TCP value, devices that support TCP on the network attached to port 1 are displayed. The following display appears:

```
NETWORK 1 MAP
0-%08000200F84A          129.213.16.206          SW/1-PCS 10000
1-%08000201136A          129.213.16.208          SW/1-PCS 10000
2-%08000200D26C          129.213.17.152          SW/1-TCP 3.0
3-%080002013E86          129.213.18.47           SW/NB-IB-2.0
4-%080002013E98          129.213.19.67           SW/NB-BR-3.0
5-%080002013E9C          129.213.18.52           SW/NB-IB-2.0
6-%0800020128B8          129.213.17.147          SW/200-TCP 3.0
7-%08000201ECD3          129.213.16.215          SW/1-PCS 1001h
```

To display the software version running on each device that supports the XNS Protocol, enter:

```
SHow -SYS NetMAP xns Long
```

Because this command does not specify a port, devices that support the XNS Protocol on the networks attached to each port of the system are displayed. The following display appears:

```
0-%080002A03C1B          NETWORK &00000001 MAP
                          SW/NB-BR-3.0
0-%080002A03C1C          NETWORK &00000000 MAP
                          SW/NB-BR-3.0
0-%080002A03C1B          NETWORK &00000003 MAP
                          SW/NB-BR-3.0
3-%080002A01339          SW/NB-BR-3.0
0-%080002A03C1B          NETWORK &00000000 MAP
                          SW/NB-BR-3.0
```

Logging Configuration Changes

The AuditLog Service sends event log messages to syslog servers to provide a history of configuration changes and other events useful in monitoring NETBuilder bridge/routers. Up to five syslog servers can receive messages from the AuditLog Service. To use the AuditLog Service you must have IP and User Datagram Protocol (UDP), and a configuration of the UNIX Syslog daemon.

The log messages provided by the AuditLog Service offer information concerning:

- User logins and listens (logouts)
- Failed login or set privilege attempts
- Successfully executed configuration commands
- Invalid SNMP community strings
- SNMP configuration changes
- File operations
- System and dial history messages
- Reboot information

AuditLog messages are sent to the syslog server(s), and logged locally in the local audit log buffer.

Sample AuditLog display messages follows

```
<188> Seq:2 Nov 30 12:39:55 Sev:5 From:EOS/SYS/EOS Msg: System Initialized and Running
<188> Seq:311 Nov 30 13:59:24 Sev:4 From:user/SYS/CONSOLE Msg: UI cmd "rb" executed
<189> Seq:107 Nov 30 14:19:25 Sev:5 From:EOS/AuditLog/EOS Msg: System Booted
<188> Seq:45 Nov 30 15:29:01 Sev:1 From:user/WEblink/HTTP 139.87.166.66 Msg: Web Link login successful
```

The message format in the local log is as follows:

<priority> Seq:SeqNumber Sev:Severity From:<Entity/Username>/Service/Source Msg:Text

The message form in the syslog server is as follows:

Date Time <Hostname/IP address> Seq:SeqNumber Sev:Severity From:<Entity/Username>)/Service/Source Msg:Text

The message includes the following fields:

| | |
|---------------------|---|
| Priority | The priority of the message. The priority is a combination of the severity level and the LocalFacility value. |
| SeqNumber | A number from 0 to 255. This is the unique identifier for the Syslog event. |
| Hostname/IP Address | The resolved host name or IP address. When displaying the log files on the syslog server, this field is prepended to each log entry. |
| Severity | The severity level in numeric form. Severity levels are as follows: <ul style="list-style-type: none"> ■ 0=Emergency ■ 1=Alert ■ 2=Critical ■ 3=Error ■ 4=Warning ■ 5=Notice ■ 6=Info ■ 7=Debug |
| <Entity/Username> | The entity or username that initiated the log message. Username specifies the user who initiated the command. The entity EOS specifies the EOS system has initiated the log message.. |
| Service | Service of the EOS that initiated the log message. |

| | |
|--------|--|
| Source | Source of the log message. Possible sources include: <ul style="list-style-type: none"> ■ CONSOLE — The console port. ■ EOS — The Enterprise OS system. ■ Telnet <i>xxx.xxx.xxx.xxx</i> — The Telnet session IP address. ■ HTTP <i>xxx.xxx.xxx.xxx</i> — The Web Link session IP address. ■ LoadConifg — The UI LoadConfig command. This source is visible only on locally logged messages. ■ <i>xxx.xxx.xxx.xxx</i> — The IP address of the SNMP management station that initiated an SNMP SET request. This source is visible only on locally logged messages. |
| Text | A description of the event. |

For more information on Syslog messages, see the Syslog Messages appendix.

Configuring Multiple Syslog Servers

You can configure up to five syslog servers. The first syslog server is considered the “initial” server, and is configured using:

```
SETDefault -AuditLog LogServerAddr = <ip address>
```

The initial syslog server uses the local facility level and default behavior that are specified by the LocalFacility and DefAction parameters. These parameters are specified using:

```
SETDefault -AuditLog LocalFacility = <0-7>
SETDefault -AuditLog DefAction = Include | Exclude
```

Include indicates that messages are logged to the Syslog. Exclude indicates that messages are not logged to the Syslog.

Additional syslog servers are added using:

```
ADD -AuditLog LogServerAddr <IP address> [<local_facility> (0-7)]
[<default_action> Exclude | Include]
```

Using the ADD LogServerAddr command, you can optionally configure the local facility value and default behavior to be different from the initial syslog server.

To delete a logserver, use the command:

```
DElete -AuditLog LogServerAddr <IP address> | All
```

To display configured logservers, use the command:

```
SHow -AuditLog LogServerAddr
```

Sample out put from this command is as follows:

| Log Server | Facility | DefAction |
|-------------|----------|-------------------|
| ----- | ----- | ----- |
| 100.100.1.1 | 7 | Include (initial) |
| 100.100.1.3 | 3 | Exclude |
| 100.100.1.4 | 2 | Include |

Managing AuditLog Filters

This section describes how to create and delete AuditLog filters. You can configure the AuditLog Service to filter based on syslog server address, service, severity, facility, and/or message identifier.

When a filter is added, it takes effect immediately. AuditLog filter rules are executed in the order of their <filterid>. The filter with the lowest <filterid> is applied first, then the filter with the next lowest <filterid>, and so on.

For example, if filter 5 is an include filter, and filter 6 is an exclude filter, the message is sent.

Defining a Filter Using the ADD LogFilter Command

To define a filter, use:

```
ADD -AuditLog LogFILTER <filterid> Include | Exclude
[Logserver=<log_server_addr_list>] [SErvice=<service_list>]
[Facility=<facility_list>] [SEVerity=<severity_list>]
[Message=<message_id_list>]
```

For detailed information on rule syntax and corresponding values, see the AuditLog Service Parameters chapter in *Reference for Enterprise OS Software*.

Displaying Filters

To display all the filters that are currently defined, use:

```
SHow -AuditLog LogFILTER
```

Deleting Filters

Filters must be individually deleted from the system.

To delete a filter, use:

```
DElete -AuditLog LogFILTER <filterid>
```

The filter is deleted immediately.

To delete all filters, use:

```
DElete -AuditLog LogFILTER ALL
```

AuditLog Filter Examples

This section provides the following five examples of AuditLog filtering:

- Log all events of severity level 5 and above to the default server
- Suppress all ISDN up/down events when severity level 3 and above are sent to the default server
- Send all VPN-related Syslog messages to server 100.100.1.2 and all other Syslog messages to all servers
- Suppress sending all Syslog messages resulting from UI commands to the default server
- Send events with severity level 4 to server 100.100.1.1 and events with severity level 5 to server 100.100.1.2

Log All Events of Severity Level 5 (Notification) and Above to the Default Server

In this example, all events of severity level 5 (Notification) and above are sent to the default syslog server. Events with severity levels lower than 4 are not sent to any syslog servers.

The following commands are used to configure this example:

```
SETDefault -AuditLog CONTROL = (Config,Messages,Security)
SETDefault -AuditLog LogServerAddr = 100.100.1.1
SETDefault -AuditLog DefaultAction = Exclude
ADD -AuditLog LogFilter 1 Include Logserver=100.100.1.1
    Severity=0-5
```

Suppress All ISDN UP/DOWN Events When Severity Level 3 (Error) and Above Are Sent to the Default Server

In this example, ISDN UP/DOWN events are suppressed. Other events with severity level 3 (Error) and above be sent to the default server. Events with severity levels less than 3 are not sent.

The following commands are used to configure this example:

```
SETDefault -AuditLog CONTROL = (Config,Messages,Security)
SETDefault -AuditLog LogServerAddr = 100.100.1.1
ADD -AuditLog LogFilter 1 Include Logserver 100.100.1.1
    Severity 0-3
SETDefault -AuditLog LogFilter 2 Exclude Logserver 100.100.1.1
    Message 1309,1310
```

Send All VPN-related Syslog Messages to Server 100.100.1.2 and All Other Syslog Messages to All Servers

In this example, VPN-related events are sent to one syslog server, while all other events are sent to a different syslog server. VPN-related events are those events sent by the RAS and IPSEC services.

The filter causes messages from IPSEC and RAS to be sent to 100.100.1.2 because of the default action, but blocked from being sent to server 100.100.1.1 because of the action specified by the filter `Exclude`. All other messages are handled by the default action, which is to send messages to both syslog servers.

The following commands are used to configure this example:

```
SETDefault -AuditLog CONTROL = (Config,Messages,Security)
SETDefault -AuditLog DefAction = Include
SETDefault -AuditLog LogServerAddr = 100.100.1.1
ADD -AuditLog LogServerAddr = 100.100.1.2
SETDefault -AuditLog LogFilter 1 Exclude Logserver 100.100.1.1
    Service IPSEC, RAS
```

Suppress Sending All Syslog Messages Resulting from UI Commands to the Default Server

In this example, events generated by entering UI commands are not sent to the default syslog server. All other types of events are sent. Events generated by entering UI commands have an identifier of 9999.

The following commands are used to configure this example:

```
SETDefault -AuditLog CONTROL = (Config,Messages,Security)
SETDefault -AuditLog LogServerAddr = 100.100.1.1
ADD -AuditLog LogFilter 1 Exclude Message=9999
```

Send Events with Severity Level 4 to Server 100.100.1.1 and Events with Severity Level 5 to Server 100.100.1.2

In this example, events with severity level 4 are sent to one syslog server and events with severity level 5 are sent to a different syslog server. All other types of events are not sent. There is no default log server.

The following commands are used to configure this example:

```
SETDefault -AuditLog CONTROL = (Config,Messages,Security)
SETDefault -AuditLog DefaultAction = Exclude
ADD -AuditLog LogServerAddr = 100.100.1.1
ADD -AuditLog LogServerAddr = 100.100.1.2
ADD -AuditLog LogFilter 1 Include Logserver=100.100.1.1 SEverity=4
ADD -AuditLog LogFilter 2 Include Logserver=100.100.1.2 SEverity=5
```

Configuring the Network Management Station for AuditLog

The AuditLog Service uses the Syslog logging service provided by the syslogd daemon available on most UNIX systems. To use AuditLog, the network must support the IP/UDP protocol between the NETBuilder II bridge/router and the network management station. Because delivery of UDP messages is not guaranteed, some Syslog messages may be lost due to network conditions between the NETBuilder II bridge/router and the network management station.

Before using the AuditLog Service, you must configure a network management station to receive NETBuilder log messages. On a UNIX network management station that already has the Syslog daemon running, follow these steps:

- 1 Log on to your network management station as root.
- 2 Add entries for local0 through local7 (or the facility selected through the SETDefault -AuditLog LocalFacility parameter) at the end of your /etc/syslog.conf file. There are two ways to do this:
 - a Use the command:


```
SETDefault -AuditLog LocalFacility = <0-7>
```

 For example:


```
SETDefault LocalFacility = 4
```
 - b Use the command:


```
ADD -AuditLog LogServerAddr <IP address>
[<LocalFacility> (0-7)] [Exclude | Include]
```

 For example:


```
ADD LogServerAddr 139.87.166.66 7
```
- 3 Create an empty log file to receive the Syslog messages. This must be the same file you specified in syslog.conf. For example, use the filename:


```
cat /dev/null > /var/log/auditlog
```
- 4 Restart the Syslog daemon by entering:

```
kill -1 `cat /etc/syslog.pid`
```

SNMP Event Notification Traps

The AuditLog Service for the NETBuilder II bridge/router provides notification for tracking and management of NETBuilder configuration changes. Notification is based on SNMP enterprise-specific trap messages. You can generate traps independently, even if the audit log capability has been disabled.

The following types of events can trigger a change or notification trap:

- Configuration Change Trap (Numeric Value 101)
This trap tracks changes originating from the execution of the SET, SETDefault, Add, and DElete commands.
- User Authentication Trap (Numeric Value 102)
This trap captures failed logins.
- File Operation Command (Numeric Value 103)
This trap captures file operations from COpy, RemoveFile, RemoveDir, ReName, MakeDir, GET, and PUT commands.

Audit log notification requires no specific configuration or parameters, but it is associated with the configuration of the SNMP Service trap generation. To configure SNMP for notification traps, follow these steps:

- 1 Enable SNMP trap generation by entering:

```
SETDefault -SNMP CONTRrol = Trap
```

- 2 Add an SNMP community for your network management station using:

```
ADD -SNMP COMMunity <"name"> TRiv RW ALl
```

The value ALl specifies that all SNMP traps are enabled. The SNMP trap types are general, authentication, and enterprise-specific. The change traps are enterprise-specific. You can also specify an SNMP trap profile, and use the AUDitLog keyword. For more information, see the SNMP Service Parameters chapter in *Reference for Enterprise OS Software*.

- 3 Specify a destination for traps using:

```
ADD -SNMP MANager <community> <IPAddress>[<mask>]
```

Remote Access of Your System

You can access your system remotely to perform network management operations. These operations can be completed through a remote station, such as a Sun workstation, which does not need to be physically connected to the console port of your system. Remote access is accomplished by using the proprietary REMote command from another 3Com bridge/router or by using the Telnet Protocol from a Telnet client.

Depending on your reasons for accessing the system remotely, you may want to use TELnet instead of REMote. These commands differ in the following ways:

- The REMote command provides access to a subset of the bridge/router commands, is UDP-based, and can truncate long displays.
- The TELnet command provides access to all bridge/router commands and is TCP-based.

After the connection has been established through remote mode or the Telnet Protocol, you can change your privilege level to Network Manager, provide the correct password, and perform network management operations or configuration procedures.

You can also prevent unauthorized users from making remote connections to your system by configuring the `-SYS NetAccess` parameter, and you can restrict remote access to specific users by configuring the `-SYS RemoteManager` parameter.

Using the REMote Command or the TELnet Command

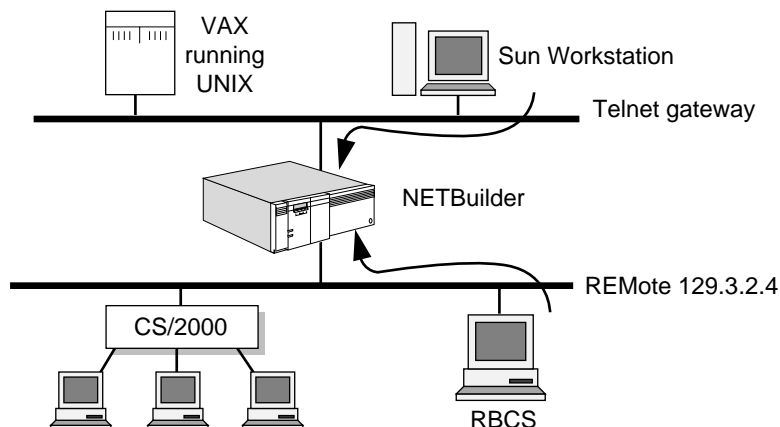
The REMote command allows you to execute commands on your bridge/router from a remote terminal. After you enter the REMote command followed by an IP address or name, you are in remote mode. Remote mode is indicated on your remote terminal by the appearance of the remote prompt (Remote:). In remote mode, all commands entered affect your bridge/router until you exit remote mode. To exit remote mode, press the Break key and enter TELnet to return to Telnet mode.



The REMote command is not subject to a configured password. To restrict access to the system, disable REMote and use the TELnet command instead.

To display the software version on a bridge/router with the address 129.3.4.2, see Figure 453, and follow these steps.

Figure 453 Accessing the System Remotely



- 1 On the remote RBCS terminal, enter:

```
REMOte 129.3.4.2
```

The prompt from the remote system is displayed.

- 2 At the remote prompt, enter:

```
SHOW -SYS VERSion
```

The version information of the software running on the system is displayed.

- 3 Press the Break key to return to the command prompt of your remote terminal.

You also can enter the REMote command, followed by the address of the bridge/router, and then followed by a command to be executed.

For example, the following command displays the IP Routing Table of the system that has the address of 129.3.2.4:

```
REMOte 129.3.2.4 SHOW -IP AllRoutes
```

Some bridge/router commands cannot be used in remote mode. For a list of these commands, see the Commands chapter in *Reference for Enterprise OS Software*.

The Telnet Protocol also can be used on a remote terminal to access your bridge/router. In this situation, your system functions as the Telnet server (the destination), and the remote terminal functions as the Telnet client (the initiator).

To access a bridge/router called "gateway," on the remote terminal (RBCS or Sun workstation), see Figure 453 and enter:

```
TELnet gateway
```

The user level command prompt of the bridge/router named "gateway" appears on the remote terminal. After you change the privilege level to Network Manager and enter the password, you can perform network management procedures. After your management activities are complete, enter the Llisten command to disconnect the session and place the port in listen mode.

Preventing Remote Access

By default, your system can be accessed remotely using the REMote command or the Telnet Protocol or via the WEB browser. You can regulate access from remote devices by using the SETDefault -SYS NetAccess command.

- To disable access with the REMote command, enter:

```
SETDefault -SYS NetAccess = NoRemote
```

- To disable access with the Telnet Protocol, enter:

```
SETDefault -SYS NetAccess = NoTelnet
```

- To disable access with the WEB browser, enter:

```
SETDefault -SYS NetAccess = NoWeb
```



CAUTION: *The software allows the bridge/router to be disabled without giving any warning messages. After assigning NoRemote or NoTelnet, you can no longer access the system parameters to perform software configuration. You must boot the system software diskette that contains an enabled NetAccess parameter before you can regain access.*

If IP security options are implemented on the system ports, a remote station without matching IP security options is not allowed to access the system. For information on restricting access to IP routers and end system configurations, see the IP Security Options chapter.

Restricting Remote Access

To specify that only certain devices can access your system, use:

```
ADD -SYS RemoteManager <IPAddress>
```

This command does not control XNS-based remote access.

To allow only the device with the address 129.98.96.99 to access your system, obtain Network Manager privilege and follow these steps:

- 1 Remove remote access to your system from all devices by entering:
- 2 Specify that only the address 129.98.96.99 can access your system by entering:

```
DELEte -SYS RemoteManager *.*.*.*
```

```
ADD -SYS RemoteManager 129.98.96.99
```

You can configure a maximum of three RemoteManager addresses. For additional information on these commands, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*.

Restricting Telnet Access

You can specify that only certain devices can access your system using the Telnet Protocol using:

```
ADD -SYS TelnetManager <IP address>
```

For example, to allow only the devices with the address 129.213.48.18 and 139.87.180.* to access your system using the Telnet Protocol, obtain Network Manager privilege and complete the following steps:

- 1 Remove Telnet access to your system from all devices by entering:
- 2 Specify that only the devices with the address 129.213.48.18 and 139.87.180.* can access your system by entering:

```
DELEte -SYS TelnetManager *.*.*.*
```

```
ADD -SYS TelnetManager 129.213.48.18
```

```
ADD -SYS TelnetManager 139.87.180.*
```

You can configure a maximum of six TelnetManager addresses. For more information, see the SYS Service Parameters chapter in *Reference for Enterprise OS Software*.

Resynchronization Feature for Encryption Devices

You can connect KG-81/KG-94 encryption devices to WAN ports (for example, ports 3 and 4, which have an RS-449 connector). The KG-81 and KG-94 encryptors are available for use by U.S. government installations only.

After you establish the connection between the bridge/router and one of these devices, enable the resynchronization feature on the bridge/router using:

```
SETDefault !<path> -PATH CONTrol = Crypto
```

For example, to enable the resynchronization feature on path 3, enter:

```
SETDefault !3 -PATH CONTrol = Crypto
```

When this feature is enabled, if a path goes down, a pulse that attempts to resynchronize the devices is generated on one of the signal lines (DTR) on the RS-449 connector. Each pulse is 10,000 microseconds long and is sent approximately every 10 seconds. Pulses will be sent until resynchronization occurs.

Disable the resynchronization feature only when you disconnect the bridge/router from the device. In the following example, the resynchronization feature is being disabled on path 3:

```
SETDefault !3 -PATH CONTROL = NoCrypto
```

To display the current settings of the CONTROL parameter in the PATH Service, enter:

```
SHow -PATH CONTROL
```



This feature is supported only on ports running PPP.

LAN Net Manager Support

You can configure the bridge/router to provide information to LAN Net Manager stations on your token ring network. LAN Net Manager is an IBM network management application used to monitor and perform some configuration of token ring networks. As implemented by IBM, the LAN Net Manager application communicates with one of five management servers that reside on each token ring network. These servers provide information to LAN Net Manager regarding current conditions of the ring, and provide some control over the ring. For example, using the application you can remove stations from the ring or change their operating parameters.

Of the five IBM management servers, the bridge/router provides support for the following four servers:

- LAN Reporting Mechanism
- Ring Error Monitor
- Configuration Report Server
- Ring Parameter Server

For more information, see your LAN Net Manager documentation.

Configuring LAN Net Manager Support

When you configure LAN Net Manager support on a bridge/router, the bridge/router and the attached token rings become eligible for monitoring by the IBM LAN Net Manager application.

By default, LAN Net Manager support is disabled. To configure a bridge/router for LAN Net Manager support, follow these steps:

- 1 Set the baud rate for the LAN Net Manager paths using:

```
SETDefault !<path> -PATH BAud = <kbps>
```

- 2 Enable global bridging by entering:

```
SETDefault -BRIDGE CONTROL = Bridge
```

- 3 Enable Logical Link Control, type 2 (LLC2) on the LAN Net Manager ports using:

```
SETDefault !<port> -LLC2 CONTROL = Enable
```


- 4 Configure source route bridging.

- a Assign a unique bridge number to the bridge/router using:

```
SETDefault -SR BridgeNumber = <number> (0-15) | 0x<number> (0-F)
```

- b Assign a unique ring number to each LAN Net Manager port:

```
SETDefault !<port> -SR RingNumber = <number> (1-4095) | 0x<number> (1-FFF)]
```

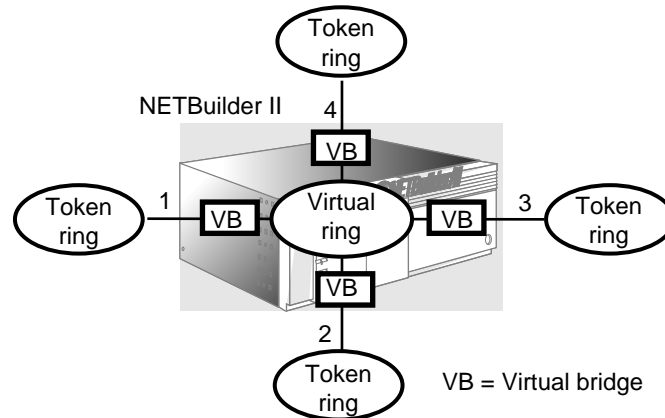
- c Configure the LAN Net Manager ports for source route bridging using:
- ```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```
- d Turn on source route discovery for LLC2 on the LAN Net Manager ports using:
- ```
SETDefault !<port> -SR RouteDiscovery = LLC2
```
- 5 Reenable the LAN Net Manager paths and ports using:
- ```
SETDefault !<path> -PATH CONTROL = Enabled
SETDefault !<port> -PORT CONTROL = Enabled
```
- 6 If LAN Net Manager is used in Virtual Ring mode, set up a virtual ring and bridges.
- a Assign a virtual ring number to the bridge/router using:
- ```
SETDefault -LNM VirRingNum = <number> (1-4095)
```
-  *Once a virtual ring number is assigned to a bridge/router, it cannot be linked to as a physical bridge.*
- b Assign a virtual bridge number to each LAN Net Manager port using:
- ```
SETDefault !<port> -LNM VirBrNum = <number> (0-15)
```
- 7 If the password for Bridge parameters in LAN NET Manager has been changed, enter the same password in the 3Com bridge/router, using:
- ```
SETDefault -LNM PassWord = "<string>"
```
- 8 Set the number of alternate LAN Net Manager stations supported by the bridge/router using:
- ```
SETDefault -LNM NumAltMgrs = <number> (0-5)
```
- The default value is 4.
- 9 Enable LAN Net Manager control by entering:
- ```
SETDefault -LNM CONTROL = Enabled
```

You can set other parameters in the LNM Service to customize timers and thresholds for your LAN Net Manager configuration. For more information about parameters in the LNM Service, see the LNM Service Parameters chapter in *Reference for Enterprise OS Software*.

Configuring Virtual Bridges and a Virtual Ring for NETBuilder II

When supporting LAN Net Manager, the NETBuilder II system must adapt to certain limitations imposed by the LAN Net Manager application. LAN Net Manager assumes that all bridges have only two ports, and as a result imposes this limit on the number of ports on a bridge. Since the NETBuilder II system can support multiple bridged ports, these must appear to LAN Net Manager as multiple two-port virtual bridges. Each virtual bridge connects a token ring port to an internal virtual ring.

Figure 454 is an example of a NETBuilder II system with four virtual bridges connected to a single virtual ring that is internal to the system.

Figure 454 Virtual Bridges and Virtual Ring on a NETBuilder II System

When configuring LAN Net Manager to monitor the token rings of a NETBuilder II bridge/router, several virtual bridges must be defined, one for each NETBuilder II token ring. One port of each virtual bridge corresponds to a real token ring port while the other port is attached to the virtual ring. In the figure, for example, there are four virtual bridges, each connected to a token ring.

When configuring this virtual bridge on the LAN Net Manager, enter the media access control (MAC) address of the token ring port and any dummy MAC address you may have configured for the virtual port. The NETBuilder II system automatically assumes it is a port on the virtual ring. On the NETBuilder II system, bridge numbers must be assigned to each of the virtual bridges, and a unique ring number must be assigned to the virtual ring. These numbers are used only to work around the two port limitations of the LAN Net Manager, and will not affect other source route bridging operations.

Disabling LAN Net Manager Support

To disable LAN Net Manager support, enter:

```
SETDefault -LNM CONTROL = Disable
```

When LAN Net Manager support is disabled, the bridge/router does not respond to requests from LAN Net Manager stations, nor does it send notifications to LAN Net Manager stations. If LAN Net Manager support is disabled when reporting links to LAN Net Manager stations are established, links will be gracefully terminated (as defined by IBM) by the bridge/router before disabling.

To reenable LAN Net Manager support, enter:

```
SETDefault -LNM CONTROL = Enable
```

If you enable LAN Net Manager support on a token ring where LAN Net Manager is not resident, you must configure the bridge/router to use end system source routing. To do this, turn on source route discovery for LLC2 using:

```
SETDefault !<port> -SR RouteDiscovery = LLC2
```

AMP-Based Network Device Discovery

Adapter Management Protocol (AMP) discovery is a 3Com protocol used by 3Com network management platforms to discover network devices attached to LAN segments. AMP operates at the MAC/LLC layer and uses group addressing. AMP discovery to 3Com bridge/routers includes a built-in discovery responder in the software.

Discovery is accomplished by a station transmitting a discovery request frame addressed to the AMP group address. If the transmitting station is attached to an Ethernet or FDDI segment, a multicast address is used. If the station is attached to a token ring segment, a functional address is used. Devices receiving the request frame respond by transmitting a discovery response frame directly addressed (unicast) to the requesting station.

The responder for the bridge/router listens for discovery request frames on LAN media interfaces (Ethernet, FDDI, token ring, and bridged serial). A bridge/router operating as both a bridge and a router responds to and forwards requests over bridged interfaces. When a request frame is forwarded between an Ethernet or FDDI segment and a token ring segment, the destination address requires a mapping between multicast and functional addresses. Only request frames require this mapping; all other AMP discovery frames are directly addressed.

The multicast address and default functional address used for AMP discovery are shown in Table 112. The multicast address is reserved and is not configurable. The functional address is configurable.

Table 112 AMP Multicast and Functional Addresses

| | Noncanonical | Canonical |
|------------------------|----------------|----------------|
| AMP Multicast Address | 8006 3188 8858 | 0160 8C11 111A |
| AMP Functional Address | C000 0100 0000 | 0300 8000 0000 |

Configuring the Discovery Responder

The Discovery Responder is a built-in service, and no configuration commands apply directly to it. When the bridge/router starts up, it attempts to map the AMP multicast address to a functional address (either the AMP default or a user-defined one). If this attempt is successful, the Discovery Responder for the bridge/router will be able to receive requests that originate on a token ring segment.

User configuration is required only when the AMP default functional address needs to be mapped to some other multicast address. In this case, you can use the BRidge Service to establish a mapping between the AMP multicast address and a different AMP functional address.



CAUTION: *The functional address used for AMP discovery is a network-wide address. All devices supporting AMP discovery (including PCs with 3Com token ring adapters and bridge/routers) must use the same functional address.*

Configuring AMP Using the BRidge Service

To see which functional addresses are mapped to which multicast addresses, use the SHow command. In the following example, entry 1 shows the default AMP functional-multicast mapping.

Enter:

```
SHow -BRidge FunctionalAddr
```

The following display appears:

| No. | Functional Address | Multicast Address |
|------|--------------------|-------------------|
| ---- | ----- | ----- |
| 1 | %030080000000 | %01608C11111A |
| 2 | %0300FFFFFFFF | %FFFFFFFFFFFF |

```
      3      %030000008000      %0180C2000000
      4      %FFFFFFFFFFFFFF      %FFFFFFFFFFFFFF
-- Entries displayed = 4
```

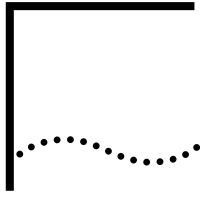
To remove the mapping of the AMP default functional address to the AMP multicast address, use the **DELeTe** command. For example:

```
DELeTe -BRidge FunctionalAddr = %030080000000
```

To map a new (non-default) AMP functional address to the AMP multicast address, use the **ADD** command. For example:

```
ADD -BRidge FunctionalAddr = %030040000000 MultiCastAddr = %01608C11111A
```

For information about available functional addresses, see “Adding Functional-Address-to-Multicast-Address Mappings to the Default Table” in the Configuring Bridging chapter.



SCHEDULING AND EVENT-BASED MACRO EXECUTION

This chapter describes how to schedule repetitive events to occur on specific days or dates using the SCheduling Service and how to set up an automatic back-up for a port using Event-based Macros Execution (EBME).



For conceptual information, see “How the Scheduler Works” and “How EBME Works” later in this chapter.

Creating Schedules

This section describes how to create schedules.

Defining a Daily Schedule

You must define at least one daily schedule before you can create an active schedule.

Create a daily schedule and define its first event in one step using:

```
ADD -SCH EVent <daily schedule> <hh:mm> <command-string>
```

You can add additional events to the daily schedule using the same syntax, where <daily schedule> is the name of a daily schedule for which at least one step has already been defined.

Creating an Active Schedule

After the daily schedule is complete, you must assign it to a calendar date or to a day of the week using:

```
ADD -SCH ActiveSChedule <mm/dd | SUN | MON | TUE | WED | THU | FRI | SAT >  
<daily schedule>
```

The scheduler is only active when two conditions are met:

- The -SCH CONTrol parameter must be set to Enabled.
- The -SCH CONTrol parameter must be set to RealTimeClock.

The CONTrol parameter also allows you to choose the hardware clock or software clock and select or deselect the logging function. For a complete summary of settings for the CONTrol parameter, see the SCH Service Parameters chapter in *Reference for Enterprise OS Software*.

Enable the scheduler by entering:

```
SETDefault -SCH CONTrol = Enabled
```

Executing Macros Using the Scheduler

The scheduler is a batch-oriented utility. Always test macros thoroughly before submitting them to the scheduler. The scheduler submits the macro to the system without evaluating it and does not report success or failure (even if logging is enabled, the scheduler reports only that the macro was submitted).

Be sure that any macros submitted to the scheduler contain:

- No infinite loops.
- No input variables.
- No illegal commands.
- Minimal output message.



CAUTION: *The only way to stop a macro that contains an infinite loop is to reboot the system.*

Scheduling WAN Connections

You can create a daily schedule to establish a dial-up connection and hang up the line at a specified times, and assign this daily schedule to one or more calendar dates or days of the week.

To use scheduled dial-up, follow these steps:

- 1 To create a daily schedule that establishes a dial-up connection, use:

```
ADD -SCH EVENT <daily schedule> <hh:mm> DIal !<port> "<telephone number>"
```

For example, to define the daily schedule, "Daily," that establishes a connection to the telephone number 555-1212 on port 3 at 11 a.m., enter:

```
ADD -SC EVENT daily 11:00 DIal !3 "5551212"
```

- 2 To add to the daily schedule an event that hangs up the connection, use:

```
ADD -SCH EVENT <daily schedule> <hh:mm> HangUp !<port>
```

For example, to add to the daily schedule, "Daily," a hangup event at 3 p.m., enter:

```
ADD -SCH EVENT daily 15:00 HangUp !3
```

- 3 To assign the daily schedule, "Daily," to each weekday, enter:

```
ADD -SCH ActiveSchedule MON DAILY
```

Repeat this step four times, substituting for "MON" the remaining weekday designators: "TUE," "WED," "THU," and "FRI."

Executing Event-based Commands/Macros

EBME provides automatic back-up if a connection fails between two sites when a primary link goes down. EBME also provides loopback detection and recovery. This is a port-based service.



For conceptual information, see "How EBME Works" later in this chapter.

EBME provides you with the following features:

- User-defined command or macro configuration.
- A backup action that is executed when the status of a port changes. EBME can back up any kind of port including static, virtual, parent, and dial ports.
- An action that is executed when a port is in a loopback condition. EBME can disable the port and then re-enable the port after a defined delay.
- A user-configurable debounce timer that provides a delay before the command or macro is executed. This timer is provided to prevent the software from reacting to transient changes in the port status.
- A log to track system command or macro execution.
- User control to enable or disable this service.

EBME is also a batch-oriented utility. Always test macros thoroughly before submitting them to EBME. EBME submits the macro to the system without evaluating it and does not report success or failure (even if logging is enabled, the scheduler reports only that the macro was submitted).

Be sure that any macros submitted to the EBME contain:

- No infinite loops.
- No input variables.
- No illegal commands.
- Minimal output message.



CAUTION: *The only way to stop a macro that contains an infinite loop is to reboot the system.*

Setting Up a Backup Port

To configure EBME to bring up a port when a primary link fails, follow these steps:

- 1 Configure the command or macro for the port being backed up by entering:

```
ADD !2 -SCH EbmeEvent PortDown 30 DO port2down_macro
```

When port 2 fails, this command causes the EBME Service to execute the port2down_macro file and sets the debounce timer to 30 seconds. The port2down_macro brings up the backup port you specify in the macro.

- 2 Enable the EBME Service by entering:

```
SETDefault -SCH EbmeCONTROL = Enable
```

- 3 Set the Log option to record the commands or macros that are executed in the system log buffer by entering:

```
SETDefault -SCH EbmeCONTROL = Log
```

Hanging Up a Port

To configure EBME to hang up a backup port when the primary link becomes active again, enter:

```
ADD !2 -SCH EbmeEvent PortUp HangUp !3
```

When port 2 comes up, this command causes the EBME Service to execute the HangUp command for port 3.

Recovering from Port Loopback

The Spanning Tree Protocol (STP) is designed to prevent bridges from forming a loop in active paths. However, STP does not prevent data from being looped back into the port that emitted it. This situation may cause an incorrect station hop and/or broadcast storm problems.

EBME can be used to detect a port loopback condition. A user-defined command or macro can be executed when a loopback condition occurs.

To configure EBME to recover from a port loopback condition, follow these steps:

- 1 To configure EBME to recover from a loopback condition on port 2, define the port recovery macro, port2LBmacro, by entering:

```
define port2LBmacro =
  setd !2 -port CONT=Disable
  pause 300
  setd !2 -port CONT=Enable
)
```

This macro checks every five minutes (300 seconds) to see if the loopback condition is still happening. When the loopback is gone, the port is returned to the forwarding state.

- 2 Enable the Spanning Tree Protocol by entering:

```
SETDefault -STP CONTROL = Enable
```

- 3 Configure the macro for the port by entering:

```
ADD !2 -SCH EbmeEvent LoopBack DO port2LBmacro
```

- 4 Enable the EBME Service entering:

```
SETDefault -SCH EbmeCONTROL = Enable
```



CAUTION: Keep in mind that the event command string is executed sequentially. A Pause command delays another event from being executed even if the other event occurs concurrently. 3Com recommends that the Pause command be used in loopback events only.

How the Scheduler Works

The SCHEDuling Service allows you to schedule repetitive events to be executed on a specified calendar date each year or on a specified day of each week. Scheduled events can consist of commands or macros. Used with the dial-up feature, the scheduler allows you to perform many useful tasks, including:

- Matching traffic prioritization with work schedules.
- Updating configuration and test-booting off-hours.
- Allowing for time zone and work habit differences, and giving several remote sites scheduled access to a single WAN port at the central site.
- Creating a more secure Internet with remote sites having scheduled access and only central site originating calls.
- Synchronizing the mail server and WAN link to control cost of electronic mail distribution.

The scheduler allows you to define up to 12 *daily schedules* and to assign each daily schedule to one or more calendar dates or days of the week. Each daily schedule contains one or more events, each of which consists of a time of day and a command or macro to be executed.

An *active schedule* is the combination of a calendar day or day of the week with a daily schedule. On any given day, only one active schedule can be in use.

If an active schedule exists for today's calendar date (for example, March 31) and an active schedule exists for today's day of the week (for example, Wednesday), the active schedule for the calendar date takes precedence.

How EBME Works

Event-based Command/Macro Execution is a platform routing protocol and provides a backup link when the primary link between two sites fails. It also provides port loopback detection EBME supports a port-level backup to a primary link failure and loopback detection and recovery.

You can use EBME to configure commands or macros for execution when the status of a given port changes from UP to DOWN, from DOWN to UP or LOOPBACK condition. You can configure commands and macros to contain the

instructions to bring back the primary connection if an UP to DOWN event occurs on the port. You can also configure commands and macros to be executed when a loopback event occurs.

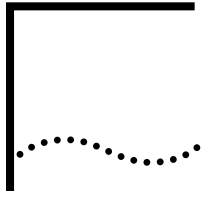
EBME consists of event generators, a port-based event responder, and an event and action database. The EBME Service provides the interface for these components and for the bridge/router software.

When the service is enabled to log all events, the sequence of events to execute the action you configured is as follows:

- When the port is configured, with some EBME events, its status is monitored by EBME.
- The port status, either up, down or loopback is recognized by the port-based event responder from the event generator. Each port may have no more than three events, up, down or loopback.
- The port up/down event becomes valid when the debounce timer runs out.
- When the event responder determines that the event has occurred, it queues the event and selects the appropriate action from the event and action database.
- The action is scheduled to take place.
- When the action has completed, a message is written to the log file stating that the action is complete.



Only the first 80 characters of the output from the UI command are printed, while only the macro name and a complete or incomplete message are printed.



SWAPPING NETBUILDER II HARDWARE MODULES

This appendix describes how to swap modules in your NETBuilder II bridge/router while your bridge/router software continues operating on other modules and ports.

You can swap one type of module with another, or you can swap two modules of the same type.

Swapping Hardware Modules

To swap hardware modules, follow these steps:

- 1 Disable the path mapped to the hardware module using:

```
SETDefault !<path> -PATH CONTROL = Disabled
```



CAUTION: When you disable the path, you risk losing network connections associated with that path.

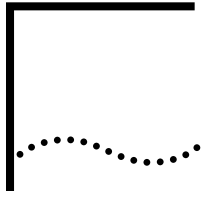
- 2 Hot-swap the board.

Remove the board and replace it with a new board.

For more information on performing the swap operation, refer to the appropriate hardware module installation guide.

- 3 Re-enable the path using:

```
SETDefault !<path> -PATH CONTROL = Enabled
```

DIAL-UP PROGRESS AND ERROR MESSAGES

This appendix provides dial-up progress and error messages for modems and integrated services digital network (ISDN) terminal adapters (TAs). It also provides information about the NETBuilder II I/O module, which supports data terminal ready (DTR) and V.25bis dialing. In addition, it lists the transmit and receive states the data terminal equipment (DTE) connector on SuperStack II NETBuilder bridge/router needs to be in to operate.



For more information about cables, modems, TAs, and telco services, see the WAN Cabling and Connectivity Guide. You can find this guide on the 3Com Corporation World Wide Web site by entering:

<http://www.3com.com/>

HSS Line Driver Cards

The NETBuilder II bridge/router supports all HSS cards except Rev. A on the HSS V.35/RS-232 module.

To verify that your HSS I/O module is not Rev. A (assembly number 06-107-000), enter:

```
SHow -SYS IOboardInfo
```

DTE Connector Transmit and Receive States

Table 113 lists each DTE connector on a NETBuilder II bridge/router and the state the connector needs to be in while data is received or transmitted.

Table 113 DTE Connector Transmit and Receive States

| State | Signal | RS-232 Pin | RS-449 Pin | V.35 | Signal Direction |
|-------|--------|------------|------------|------|--------------------|
| High | DSR | 6 | 11, 29 | E | To NETBuilder II |
| High | DCD | 8 | 13, 31 | F | To NETBuilder II |
| High | DTR | 20 | 12, 30 | H | From NETBuilder II |

Dial-Up Progress and Error Messages

The modem, model 42x and 52x SuperStack II NETBuilder bridge/routers, ISDN TA, or Enterprise OS software may return a message indicating the reason for a call failure, the progress of a call, or the presence of an incoming call.

Software Messages for Modems

The NETBuilder II bridge/router sends the following messages to indicate call progress or failure on lines configured with modems:

```
INCOMING CALL ON PATH <path>, PORT <port>  
CALL ON PATH <path>, PORT <port> REJECTED  
INCOMING CALL ON PATH <path>, PORT <port> CONNECTED  
CALL ON PATH <path>, PORT <port> CONNECTED  
CALL ON PATH <path>, PORT <port> REJECTED, NO CARRIER  
DISCONNECT ON PATH <path>, PORT <port>
```

```

PATH NOT COMING UP, INITIATING HANGUP ON PATH <path>
PRIMARY IS UP, INITIATING HANGUP ON PATH <path>
SECONDARY IS IDLE, INITIATING HANGUP ON PATH <path>
PATH IS IDLE, INITIATING HANGUP ON PATH <path>
AUTODIAL INITIATING CALL ON PATH <path>, PORT <port>
USER INITIATING CALL ON PATH <path>, PORT <port>
BOD FEATURE INITIATING CALL ON PATH <path>, PORT <port>
DR FEATURE INITIATING CALL ON PATH <path>, PORT <port>
RETRY INITIATING CALL ON PATH <path>, PORT <port>
DIALNO IS REQUIRED FOR V.25BIS CALLS ON PATH <path>
NO CALL ATTEMPTED, NO-ORIGINATE SET ON PATH <path>
CALL ON PATH <path>, PORT <port> REJECTED, CODE = <xx>
DOD INITIATING CALL ON PATH <path>, PORT <port>
DOD RETRY INITIATING CALL ON PATH <path>, PORT <port>

```



Some of these messages include a two-letter response code. This code is also displayed by the SHow -PORT DialHistory command. Table 114 lists these modem response codes.

V.25 Modems

In response to a failed attempt, a V.25bis modem may return one of the error codes listed in Table 114. These codes are associated with the following messages:

```

DIALNO IS REQUIRED FOR V.25BIS CALLS ON PATH <path>
CALL ON PATH <path>, PORT <port> REJECTED, CODE = <xx>

```

Table 114 Modem Response Codes

| Response Code | Meaning |
|---------------|--------------------------|
| AB | Abort call |
| CB | Local DCE busy |
| ET | Engaged Tone |
| FC | Forbidden call |
| NS | Number not stored |
| NT | Answer tone not detected |
| RT | Ring tone |

Software Messages for SuperStack II NETBuilder Bridge/Router

Table 115 provides error codes and messages the model 42x and 52x SuperStack II NETBuilder bridge/routers may return in response to failed attempts to communicate.

Table 115 ISDN Dial Failure Cause Codes

| Cause Number* | Cause | Definition† |
|---------------|---------------------------------------|---|
| 1 | Unallocated (unassigned) number | Indicates that the destination requested by the calling user cannot be reached because, although the number is in a valid format, it is not currently assigned (allocated). |
| 2 | No route to specified transit network | Indicates that the equipment sending this cause has received a request to route the call through a particular transit network, which it does not recognize. The equipment sending this cause does not recognize the transit network either because the transit network does not exist or because that particular transit network, while it does exist, does not service the equipment that is sending this cause. This cause is supported on a network-dependent basis. |

(continued)

Table 115 ISDN Dial Failure Cause Codes (continued)

| Cause Number* | Cause | Definition† |
|----------------------|--|---|
| 3 | No route to destination | Indicates that the called user cannot be reached because the network through which the call has been routed does not serve the destination desired. This cause is supported on a network-dependent basis. |
| 6 | Channel unacceptable | Indicates the channel most recently identified is not acceptable to the sending entity for use in this call. |
| 7 | Call awarded and being delivered in an established channel | Indicates that the user has been awarded the incoming call, and that the incoming call is being connected to a channel already established to that user for similar calls. |
| 16 | Normal call clearing | Indicates that the call is being cleared because one of the users involved in the call has requested that the call be cleared. Under normal situations, the source of this cause is not the network. |
| 17 | User busy | Used when the called user has indicated the inability to accept another call. The user equipment is compatible with the call. |
| 18 | No user responding | Used when a user does not respond to a call establishment message with either an alerting or connect indication within the prescribed period of time allocated. |
| 19 | No answer from user (user alerted) | Used when a user has provided an alerting indication but has not provided a connect indication within a prescribed period of time. |
| 21 | Call rejected | Indicates that the equipment sending this cause does not wish to accept this call, although it could have accepted the call because the equipment sending this cause is neither busy nor incompatible. |
| 22 | Number changed | Returned to a calling user when the called party number indicated by the calling user is no longer assigned. If a network does not support this capability, cause #1 "unallocated (unassigned) number" shall be used. |
| 26 | Non-selected user clearing | Indicates that the user has not been awarded the incoming call. |
| 27 | Destination out of order | Indicates that the destination indicated by the user cannot be reached because the interface to the destination is not functioning correctly. The term "not functioning correctly" indicates that a signalling message was unable to be delivered to the remote user; for example, a physical layer or data link layer failure at the remote user, user equipment off-line, and so forth. |
| 28 | Invalid number format | Indicates that the called user cannot be reached because the called party number is not in a valid format or is not complete. |
| 29 | Facility rejected | Returned when a facility requested by the user cannot be provided by the network. |
| 31 | Normal, unspecified | Used to report a normal event only when no other cause in the normal class applies. |
| 34 | No circuit/channel available | Indicates that there is no circuit/channel presently available to handle the call. |
| 38 | Network out of order | Indicates that the network is not functioning correctly and that the condition is likely to last a relatively long period of time; for example, immediately retrying the call is not likely to be successful. |
| 41 | Temporary failure | Indicates that the network is not functioning correctly and that the condition is not likely to last a long period of time; for example, the user can try another call attempt almost immediately. |
| 42 | Switching equipment congestion | Indicates that the switching equipment generating this cause is experiencing a period of high traffic. |
| 43 | Access information discarded | Indicates that the network could not deliver access information to the remote user as requested; for example, a user-to-user information, low layer compatibility, high layer compatibility, or subaddress. |
| 44 | Requested circuit/channel not available | Returned when the circuit or channel indicated by the requesting entity cannot be provided by the other side of the interface. |
| 47 | Resources unavailable, unspecified | Used to report a resource unavailable event only when no other cause in the resource unavailable class applies. |

(continued)

Table 115 ISDN Dial Failure Cause Codes (continued)

| Cause Number* | Cause | Definition† |
|----------------------|--|---|
| 49 | Quality of service not available | Used to report that the requested quality of service, as defined in CCITT Recommendation X.213, cannot be provided (for example, throughput or transit delay cannot be supported). |
| 50 | Requested facility not subscribed | Indicates that the requested supplementary service could not be provided by the network because the user has not completed the necessary administrative arrangements with its supporting network. |
| 57 | Bearer capability not authorized | Indicates that the user has requested a bearer capability that is implemented by the equipment, which generated this cause, but the user is not authorized to use. |
| 58 | Bearer capability not presently available | Indicates that the user has requested a bearer capability that is implemented by the equipment, which generated this cause, but which is not available at this time. |
| 63 | Service or option not available, unspecified | Used to report a service or option not available event only when no other cause in the service or option not available class applies. |
| 65 | Bearer capability not implemented | Indicates that the equipment sending this cause does not support the bearer capability requested. |
| 66 | Channel type not implemented | Indicates that the equipment sending this cause does not support the channel type requested. |
| 69 | Requested facility not implemented | Indicates that the equipment sending this cause does not support the requested supplementary service. |
| 70 | Only restricted digital information bearer capability is available | Indicates that an equipment has requested an unrestricted bearer service but that the equipment sending this cause only supports the restricted version of the requested bearer capability. |
| 79 | Service or option not implemented, unspecified | Used to report a service or option not implemented event only when no other cause in the service or option not implemented class applies. |
| 81 | Invalid call reference value | Indicates that the equipment sending this cause has received a message with a call reference that is not currently in use on the user-network interface. |
| 82 | Identified channel does not exist | Indicates that the equipment sending this cause has received a request to use a channel not activated on the interface for a call. For example, if a user has subscribed to those channels on a primary rate interface numbered from 1 to 12 and the user equipment or the network attempts to use channels 13 through 23, this cause is generated. |
| 83 | A suspended call exists, but this call identity does not | Indicates that a call resume has been attempted with a call identity, which differs from that in use for any presently suspended call(s). |
| 84 | Call identity in use | Indicates that the network has received a call suspend request. The call suspend request contained a call identity (including the null call identity) that is already in use for a suspended call within the domain of interfaces over which the call might be resumed. |
| 85 | No call suspended | Indicates that the network has received a call resume request. The call resume request contained a call identity information element that presently does not indicate any suspended call within the domain of interfaces over which calls may be resumed. |
| 86 | Call having the requested call identity has been cleared | Indicates that the network has received a call resume request. The call resume request contained a call identity information element that once indicated a suspended call; however, that suspended call was cleared while suspended (either by network timeout or by the remote user). |
| 88 | Incompatible destination | Indicates that the equipment sending this cause has received a request to establish a call that has low layer compatibility, high layer compatibility, or other compatibility attributes (for example, data rate), which cannot be accommodated. |
| 91 | Invalid transit network selection | Indicates that a transit network identification was received that is of an incorrect format. |
| 95 | Invalid message, unspecified | Used to report an invalid message event only when no other cause in the invalid message class applies. Used to report a resource unavailable event only when no other cause in the resource unavailable class applies. Used to report a resource unavailable event only when no other cause in the resource unavailable class applies. |

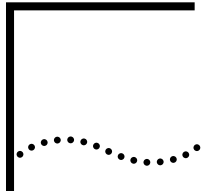
(continued)

Table 115 ISDN Dial Failure Cause Codes (continued)

| Cause Number* | Cause | Definition† |
|---------------|--|---|
| 96 | Mandatory information element is missing | Indicates that the equipment sending this cause has received a message that is missing an information element, which must be present in the message before that message can be processed. |
| 97 | Message type non-existent or not implemented | Indicates that the equipment sending this cause has received a message with a message type it does not recognize either because this is a message not defined or defined but not implemented by the equipment sending this cause. |
| 98 | Message not compatible with call state or message type non-existent or not implemented | Indicates that the equipment sending this cause has received a message such that the procedures do not indicate that this is a permissible message to receive while in the call state, or a STATUS message was received indicating an incompatible call state. |
| 99 | Information element non-existent or not implemented | Indicates that the equipment sending this cause has received a message, which includes information elements not recognized because the information element identifier is not defined or it is defined but not implemented by the equipment sending the cause. However, the information element is not required to be present in the message in order for the equipment sending the cause to process the message. |
| 100 | Invalid information element contents | Indicates that the equipment sending this cause has received an information element that it has implemented; however, one or more of the fields in the information element are coded in such a way that has not been implemented by the equipment sending this cause. One reason this cause may occur is that the bridge/router requested 56K rate adaption in auto-rate mode, but the switch does not support 56K rate adaption. |
| 101 | Message not compatible with call state | Indicates that a message has been received, which is incompatible with the call state. |
| 102 | Recovery on timer expiry | Indicates that a procedure has been initiated by the expiry of a timer in association with ETS 300 102-1 error handling procedures. |
| 111 | Protocol error, unspecified | Used to report a protocol error event only when no other cause in the protocol error class applies. |
| 127 | Interworking, unspecified | Indicates that there has been interworking with a network that does not provide causes for the actions it takes; the exact cause for a message that is being sent cannot be determined. |

* These cause numbers and definitions are found in ETS300 102-1, CCITT Q931 specification.

† In response to the error code, the dial-up software may select a new path, may try another phone number, or may keep the same path and number when retrying the call.



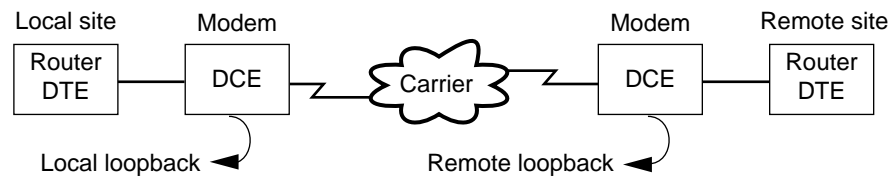
LOOPBACK TESTING

Very few modems or terminal adapters support loopback initiated by controlling the LL and RL pins on the data terminal equipment (DTE)/data communications equipment (DCE) interface. These loopback functions can be accomplished by setting the DLTest TestMode to Loopback and manually configuring the modems for the desired loopback type. This appendix describes how to set up a dial-up loopback test.

Dial-up Loopback Testing Using Modems

The following information describes how to perform a loopback test on a dial-up device and supplements the information currently found under the DLTest command in the the Commands chapter in *Reference for Enterprise OS Software*. Figure 455 shows a sample representation of the procedure.

Figure 455 Typical Dial or Leased Connection



To successfully troubleshoot your network, you need to determine the procedure your DCE uses to perform DCE loopback. There are several different ways DCEs use loopback. Check your DCE vendor manual, and identify the ways you can activate the DCE to perform the following functions:

- Local loopback
A loopback is made at the local DCE with the transmit data being returned on the receive data to the DTE (router).
- Call or connection
A connection is made to a remote DCE.
- Remote loopback
A loopback is made by a far-end DCE where the received data is being returned on the transmit path, back to the near-end DCE.

A DCE can be configured in one or more of the following ways:

- Front panel switches
- Console control through an asynchronous terminal connected to a separate control port
- Signal line control of V.24 circuits

- AT commands sent over a combination data and control port

Most DCEs do not support all of these modes of operation. To make sure which type of control your DCE offers, see your DCE vendor's documentation.

For the following information, see the Commands chapter in *Reference for Enterprise OS Software*.

- DLTest command
- High-speed serial (HSS) software settings
- Internal clock setting
- Leased-line setting

For specific settings, see the DCE manufacturer's documentation.

Before conducting a loopback test, check the basic network connectivity by following these steps:

- 1 Check the connections between the DCE and the telephone network.
- 2 Verify both DCE configuration settings.

Both DCEs should support common communication standards.

If you still experience problems, troubleshoot the connection to isolate the problem.

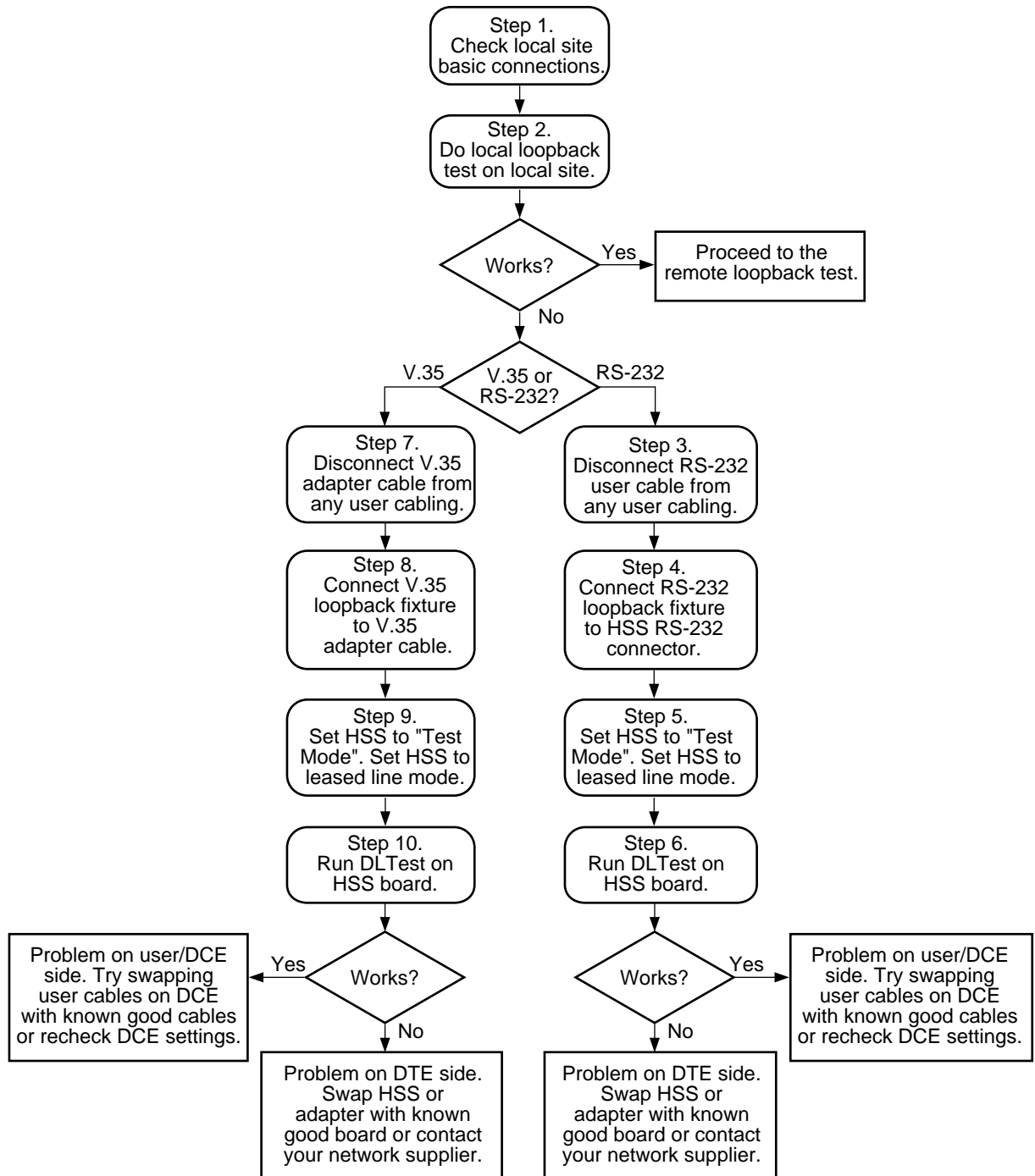


When performing the following tests, use commands or methods that are specific to each DCE.

Performing a Local Loopback Test

To conduct a local loopback test, follow these steps (see Figure 456):

Figure 456 HSS V.35 and RS-232 Local Loopback Flowchart



- 1 Check local site basic configurations.
- 2 At the local DCE, initiate the loopback mode.
 - a Set the HSS board to Leased mode.
 - b Run the loopback mode on the HSS board using the DLTest command.

If this test passes, go to the remote loopback procedure in the next section.

If the loopback test fails and you are using a V.35 interface, go to step 7.

If the loopback test fails and you are using an RS-232 interface, go to step 3.

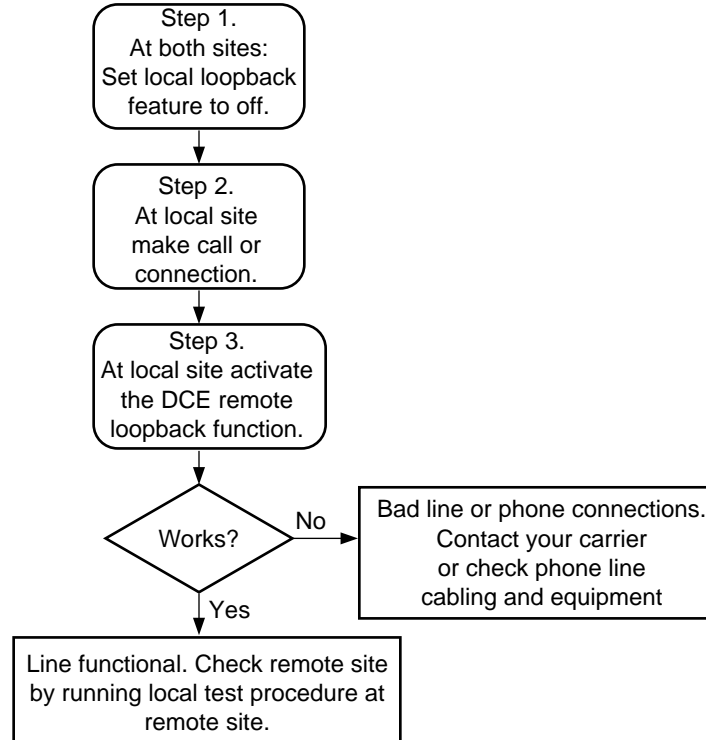
- 3 Disconnect the RS-232 user cable from the HSS board.
- 4 Connect the RS-232 loopback fixture to the HSS board (see Table 116 and Table 117).
- 5 Set the HSS board to internal clock source and leased-line mode.
- 6 Run the loopback mode on the HSS board using the DLTest command.
If the test passes, the DCE or the user cables are faulty. Try using a known good DCE or user cable. After you isolate the connection problem, remember to change back the HSS board settings.
If the test fails, try using a known good HSS board or contact 3Com.
- 7 Disconnect the V.35 user cable from the HSS board.
- 8 Connect the V.35 loopback fixture to the HSS board (see Table 116 and Table 117).
- 9 Set the HSS board to Test Mode and leased-line mode.
- 10 Run the loopback mode on the HSS board using the DLTest command.
If the test passes, the DCE or the user cables are faulty. Try using a known good DCE or user cable. After you isolate the connection problem, remember to change back the HSS board settings.
If the test fails, try using a known good HSS board or contact 3Com.

Performing a Remote Loopback Test

To perform a remote loopback test, follow these steps (see Figure 457). For DCE configuration information, see your vendor documentation.



This test is between modem and modem.

Figure 457 HSS V.35 and RS-232 Remote Loopback Flowchart

- 1 At both sites, turn the local loopback feature off.
- 2 At the local site, make the DCE call the remote DCE.
- 3 At the local DCE, start a remote loopback test using a self-generated pattern.

If the test passes, the line and remote DCE are functional. Run the local test procedure at the remote site. After you isolate the connection problem, remember to change back the HSS board settings.

If the test fails, the line or remote DCE, or phone connections at either the local or remote site are faulty; recheck the phone, cabling, DCE phone line settings, and/or contact your carrier.

Making the Loopback Fixture

To make the RS-232 loopback fixture, follow these steps (see Table 116):

- 1 Obtain a male RS-232 connector.
- 2 Wire the pins according to Table 116.

Table 116 RS-232 Loopback Pin Assignments

| Name | Pin | Name | Pin |
|------|-----|------|-----|
| TD | 2 | RD | 3 |
| RTS | 4 | CTS | 5 |
| DSR | 6 | DCD | 8 |
| | | DTR | 20 |
| RXC | 17 | TT | 24 |

To make the V.35 loopback fixture, follow these steps (see Table 117):

- 1 Obtain a male V.35 connector.
- 2 Wire the pins according to Table 117.

Table 117 V.35 Loopback Pin Assignments

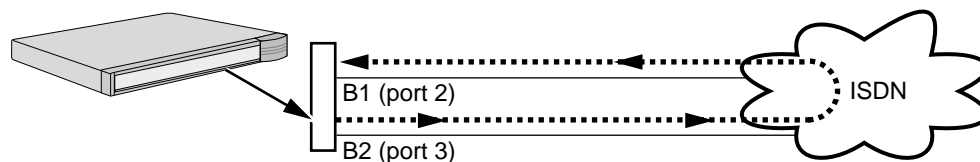
| Name | Pin | Name | Pin |
|-------|-----|------|-----|
| SCTEA | U | SCRA | V |
| SCTEB | W | SCRB | X |
| RTS | C | CTS | D |
| DTR | H | DSR | E |
| | | DCD | F |
| SCTA | Y | | |
| SCTB | AA | | |

Loopback Testing for Built-In ISDN Ports

This section describes how to perform a loopback test using two B channels on one Integrated Services Digital Network (ISDN) line.

Figure 458 shows the data flow when performing a loopback test using the two B channels of an ISDN basic rate interface (BRI) line. Both of these channels occupy the same physical connector and no modem is required in this configuration, however, the unit must be attached to an ISDN line.

Figure 458 ISDN Loopback Testing



Both channels (B1 and B2) occupy the same physical connector.

Procedure To run the loopback diagnostics test, you must have console running at 9,600 baud connected to your bridge/router.

To set up the loopback test, follow these steps:

- 1 Set the path line type to Dialup using:

```
SETDefault !<path> -PATH LineType = Dialup
```

To perform the test shown in the example, enter the following commands:

```
SETDefault !2.1 -PATH LineType = Dialup
```

```
SETDefault !2.2 -PATH LineType = Dialup
```

- 2 Set the rate adaption parameter to automatically detect the speed of the interface by using:

```
SETDefault !<path> -PATH RateAdaption = Auto
```

In the example in step 1, the test originates from path 2.2 and targets path 2.1. To specify this for path 2.2, enter:

```
SETDefault !2.2 -PATH RateAdaption = Auto
```

- 3 Set the switch type using:

```
SETDefault !<path> -PATH SwitchType = ETSI | NIT | ATT5ESS | NT1 | DMS100
|KDD
```

To set the switch type to ETSI, enter:

```
SETDefault !2 -PATH SwitchType = ETSI
```

- 4 Establish the local dial numbers for the bearer channels using the following syntax:

```
SETDefault !<port> -PATH LocalDialNo = "<string>"
```

To establish the local dial numbers for the two bearer channels, enter:

```
SETDefault !2.1 -PATH LocalDialNo = "4962124"
```

```
SETDefault !2.2 -PATH LocalDialNo = "4962125"
```

- 5 Configure the ports for loopback testing using:

```
SETDefault !<port> -PORT OWNEr = Loopback
```

Enter Loopback as the owner on both the sending and receiving ports; for example:

```
SETDefault !2 -PORT OWNEr = Loopback
```

```
SETDefault !3 -PORT OWNEr = Loopback
```

- 6 Establish a connection between the two bearer channels by dialing out on one channel and dialing into the other using:

```
Dial !<path> "<string>"
```

To dial port 2 from port 3, enter:

```
Dial !2.2 "4962124"
```

Path 2.2 places a call to the specified number, which is the number for path 2.1. It is not important which port originates or answers the call as long as the port does not try to call itself.

- 7 When the connection is successfully established, select the loopback testing mode by entering:

```
DLTest TestMode Loopback
```

You can specify the number of seconds the test should run. You can enter this value any time before entering the DLTest START command. If a value is not specified, an infinite time duration is assumed. To run the test for a specific number of seconds, use:

```
DLTest TestDuration <seconds>
```

Use caution when running the test for a specified test duration. The test ends abruptly as soon as the time duration expires, and a discrepancy between the number of packets transmitted and the number received may result.

- 8 Start the DLtest using:

```
DLTest Start <sendingport>, <receivingport>
```

To start the DLTest and designate port 2 to send the DLTest data and port 3 to receive and loop back the data, enter:

```
DLtest START 2,3
```

The loopback test is successful when the number of received packets equals or approximately equals the number of transmitted packets. If the test is not successful, verify that your system is cabled correctly and your model is installed correctly. You can check the number of packets transmitted and the number of errors by entering:

DLTest Stat

- 9 Stop the DLTest by entering:

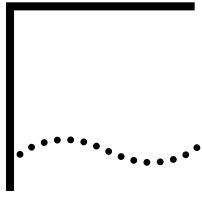
DLTest Abort

- 10 Disconnect the call by entering:

HangUp !2.2

- 11 Change the port owner from Loopback to the original owner using:

```
SETDefault !<port> -PORT OWNeR = PPP
```



INTERNET ADDRESSING

This appendix provides information about the following topics:

- Internet addresses and classes
- Dotted decimal notation
- Addressing rules
- Subnet addressing and subnet masks
- Variable length subnet masks

Internet Addresses

Any universal communications system requires a globally accepted method of identifying individual computers; one globally accepted method is to have the Network Manager assign unique Internet addresses to devices, or *hosts*, on the Internet. These hosts can be personal computers, communications servers, ports on a communications server, internetwork bridges, network control servers, or UNIX hosts. The Internet uses these assigned addresses when sending or receiving packets.



You can obtain valid and unique Internet addresses through the InterNIC Registration Services. For additional information, see the New Installation for NETBuilder II Software.

The Internet Protocol (IP) uses Internet addresses. Internet addressing uses a 32-bit address field numbered 0 to 31. This address field is composed of two parts: one part identifies the network on which the host resides, and the second part identifies the host itself. Hosts attached to the same network must share a common prefix designating their network number. Conceptually, each address is a pair (*net#*, *host#*) where *net#* identifies the network, and *host#* identifies a host on that network.

Internet addressing is divided into four classes: A, B, C, and D. Each address class begins with a unique bit pattern that is used by the Internet software residing on network hosts to identify the address class. Once the software has identified the address class from the leading bits of the Internet address, it can determine which bits are used to represent the network number and which bits are used to identify the host portion of the address. The next four sections describe the four Internet address classes.

Class A Address Format

The first type of address, Class A, has a 7-bit network field and a 24-bit local address. The highest-order bit is set to 0. This allows 127 Class A networks to be defined (network number 0 is not allowed).

The following diagram describes the format of a Class A address with the bit numbers on the first line:

| | | | | |
|---|---------|---|--------------------|----|
| 0 | 1 | 7 | 8 | 31 |
| 0 | Network | | Local Host Address | |

Class B Address Format

The second type of address, Class B, has a 14-bit network field and a 16-bit local address. The two highest-order bits are set to 1 and 0. This allows 16,383 Class B networks to be defined.

The following diagram describes the format of a Class B address with the bit numbers on the first line:

| | | | | | |
|---|---------|---|----|--------------------|----|
| 0 | 1 | 2 | 15 | 16 | 31 |
| 1 | Network | | | Local Host Address | |

Class C Address Format

The third type of address, Class C, has a 21-bit network field and an 8-bit local address. The three highest-order bits are set to 1, 1, and 0. This allows 2,097,151 Class C networks to be defined.

The following diagram describes the format of a Class C address, with the bit numbers on the first line:

| | | | | | | |
|---|---|---------|---|----|--------------------|----|
| 0 | 1 | 2 | 3 | 23 | 24 | 31 |
| 1 | 1 | Network | | | Local Host Address | |

Class D Address Format

The fourth type of address, Class D, has a 23-bit multicast address field. The four highest-order bits are set to 1, 1, 1, and 0. This allows 8,000,000 multicast addresses to be defined.

The following diagram describes the format of a Class D address, with the bit numbers on the first line:

| | | | | | | | |
|---|---|---|---|---|-------------------|---|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 31 |
| 1 | 1 | 1 | 0 | 0 | Multicast Address | | 1 |

* Not used.



No addresses are allowed with the four highest-order bits set to 1-1-1-1. These addresses are reserved.

Dotted Decimal Notation

An Internet address is specified as four decimal numbers, each separated by a dot. This format is called *dotted decimal notation*. The 32-bit Internet address is divided into four 8-bit fields, called octets; the value of each field is specified as a decimal number with the fields separated by periods.

For example, the Internet address of USC-ISIB.ARPA in binary octets and dotted decimal notation is as follows:

Binary octets: 00001010 00000011 00000000 00110100

Dotted decimal notations: 010.003.000.052 or 10.3.0.52

Valid network numbers for each address class are provided. The “nnn” represents the network portion of the address, which is assigned by the InterNIC. The “hhh” represents the host portion of the address, which is assigned by the network manager.

Class A networks: (nnn.hhh.hhh.hhh):001.hhh.hhh.hhh through 126.hhh.hhh.hhh

Class B networks: (nnn.nnn.hhh.hhh):128.001.hhh.hhh through 191.254.hhh.hhh

Class C networks: (nnn.nnn.nnn.hhh):192.000.001.hhh through 223.255.254.hhh

Class D networks: 224.000.000.000 through 239.255.255.255



The bits defining the local address portion of an Internet address cannot be all zero bits or all 1 bits. These are special addresses and are described in “Addressing Rules” next.

Addressing Rules

These general guidelines should be observed when assigning Internet addresses:

- The bits used to define the host portion of an Internet address should not be all one bits.

According to the standard, any Internet address with the host portion consisting of all ones is interpreted as meaning “all,” as in “all hosts.” For example, the address 128.1.255.255 is interpreted as meaning all hosts on network 128.1 and is reserved for directed broadcast addressing.

- The bits used to define the network portion of an Internet address should not be all zero bits.

According to the standard, any Internet address with the network portion consisting of all zeros is interpreted as meaning “this,” as in “this network.” For example, the address 0.0.0.63 is interpreted as meaning host 63 on this network.

- The class A network number 127 is assigned the “loopback” function.

The loopback function allows a datagram to be sent by a higher-level protocol to network 127 to loopback inside the host. For example, in a Berkeley Software Distribution (BSD) UNIX environment, if program A needs to communicate with program B, which is running on the same machine, they can do this with IP network number 127. However, no datagram should ever appear on any network with a source or destination network address of 127.

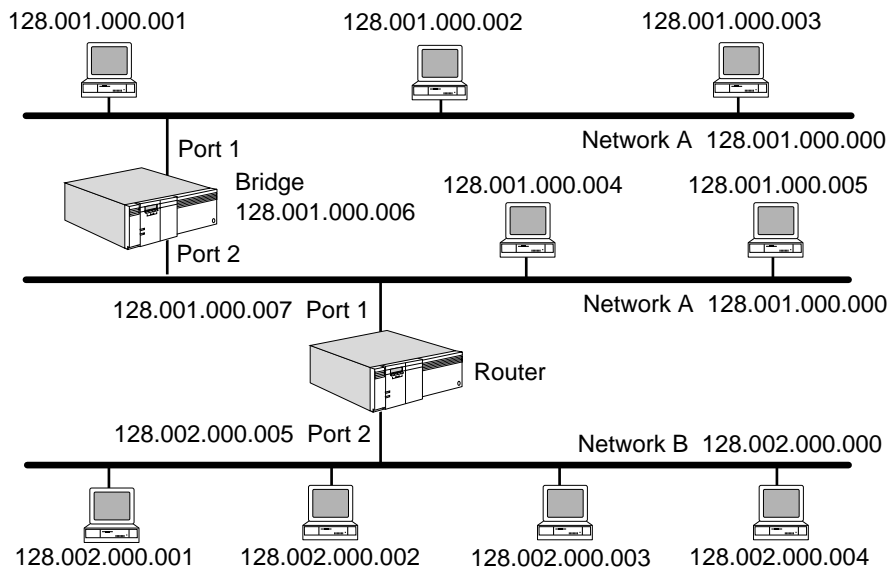
- No addresses are allowed with the four highest-order bits set to 1111. These addresses are reserved for Class E networks.

Sample Network Using the Class B Address Format

Figure 459 shows a sample network using the Class B address format. As shown in this illustration, segments connected by internetwork bridges share the same network fields while having different host fields. For example, the bridge segments Network A into two LANs, both of which have the network address of 128.001.

Segments interconnected by routers must have different network fields to be physically separate networks. For example, the router physically separates Network A from Network B as indicated by the different network addresses (Network A has a network address of 128.001 and Network B has a network address of 128.002).

Figure 459 Sample Network Using the Class B Address Format



Subnet Addresses and Subnet Masks

The original interpretation of Internet addresses (described in the previous sections) was based on a two-level hierarchy. In this model, each host sees its network as a single entity.

A number of organizations have added a third level to the interpretation of Internet addresses. In this structure, a given Internet network is divided into a collection of subnets. The three-level model is useful in networks in moderately large organizations, where it is often necessary to use more than one LAN cable to cover a local area. Each LAN can then be treated as a subnet belonging to a given main Internet network number. These independent networks are then connected by routers. However, each organization that wants to connect to the Internet can usually obtain only a single Internet number.

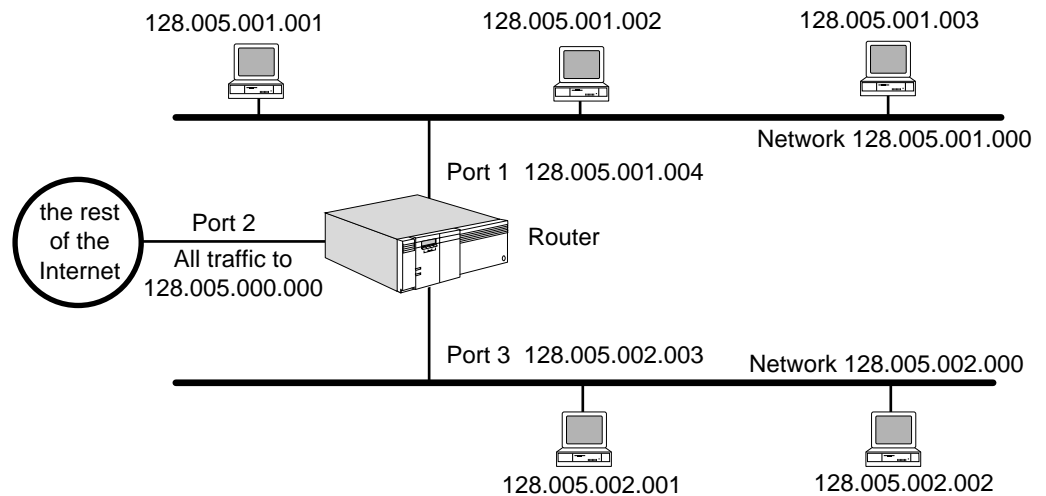
Subnet Addressing

If multiple Transmission Control Protocol/Internet Protocol (TCP/IP) networks are interconnected across routers, you must assign a different network field to each network. (However, if the network is part of the Internet, you cannot use different network fields because the network field must be assigned by the InterNIC.) Subnet addressing allows an organization to use a single Internet network number

for multiple physical networks. Subnets can be used with any class of Internet addressing except Class D.

In Figure 460, a site with two physical networks uses subnet addressing to span them with a single class B network address. The router accepts all traffic for network 128.005.000.000 and chooses a physical network based on the third octet of the address.

Figure 460 Subnet Addressing



The format of a regular Internet address and an Internet address with subnet mask are as follows:

Regular Internet Address Format

| | |
|----------------|-------------|
| Network Number | Host Number |
|----------------|-------------|

Subnet Address Format

| | | |
|----------------|---------------|-------------|
| Network Number | Subnet Number | Host Number |
|----------------|---------------|-------------|

The network field is already defined in the previous section. The width of the subnet field is constant for a given network number.

For example, on a Class B network with a 6-bit-wide subnet field, an Internet address can be broken down as follows:

| | | | | | | |
|-----|---------|----|--------|----|--------------------|----|
| 0 1 | 2 | 15 | 16 | 21 | 22 | 31 |
| 1 0 | Network | | Subnet | | Local Host Address | |



The host portion of an Internet address with the preceding definition cannot be defined as all 1 bits, but the subnet portion of an Internet address can be defined as all 1 bits.

In the preceding example, the subnet field can have any value between 0 and 63, and the host field can have any value between 1 and 1022 (all numbers are decimal). A typical class B Internet address that fits the requirements of the preceding example is the Internet address 128.5.61.100 with a subnet mask of 255.255.252.0.

Subnet Masks

The subnet mask allows the host portion of an Internet address to be divided into two parts. One part is used to identify a physical subnet, and the second part is used to identify a host on that subnet.

Bits in the subnet mask are set to 1 if the network treats the corresponding bit in the Internet address as part of the network address. Bits in the subnet mask are set to 0 if it treats the bit as part of the host identifier.

Subnet Address Format

| | | |
|----------------|---------------|-------------|
| Network Number | Subnet Number | Host Number |
|----------------|---------------|-------------|

Subnet Mask

| | | |
|-------------------|----------|----------|
| 11111111 11111111 | 11111111 | 00000000 |
|-------------------|----------|----------|

The subnet mask is also defined in the dotted decimal notation. For example, with a Class B address of 128.121.61.100, the subnet mask is as follows:

Subnet Address Format for 128.121.61.100

| | | |
|-------------------|----------|----------|
| 10000000 01111001 | 00111101 | 01100100 |
|-------------------|----------|----------|

Subnet Mask

| | | |
|-------------------|----------|----------|
| 11111111 11111111 | 11111111 | 00000000 |
|-------------------|----------|----------|

The subnet mask for the preceding example would then be 255.255.255.0.

Adhering to the preceding constant width requirement, when using RIP the value of the subnet mask should be the same on all subnets defined for a given network

number. 3Com bridge/routers support variable length subnet masks when using OSPF. For more information, see "Variable Length Subnet Masks" later in this chapter.

Subnet masks are assigned using the NETaddr parameter in the IP Service. For example, to assign the IP address of 128.005.001.001 with a subnet mask of 255.255.255.000 to port 1 of a router, enter:

```
SETDefault !1 -IP NETaddr = 128.005.001.001 255.255.255.000
```

For more information on the NETaddr parameter, see the IP Service Parameters chapter in *Reference for Enterprise OS Software*.

Subnets: Example 1

The InterNIC assigns you Class B Internet address 128.001.000.000. You need to establish 256 subnets with each subnet capable of supporting 254 hosts. This is the simplest form of subnetting. The first and second octets of the IP address identify the network, the third octet identifies the subnet, and the fourth octet identifies a host on the subnet.

To solve this problem, follow these steps:

- 1 Convert the address assigned by the InterNIC to binary format.

For example:

128.001.000.000 = 10000000 00000001 00000000 00000000

The underlined binary digits represent the network portion of the Internet address assigned by the InterNIC.

- 2 Determine the number of binary digits you need to represent 256 subnets.

Eight binary digits are required to define 256 subnets ($2^8 = 256$). The binary values of all zeros (decimal value 0) and all ones (decimal value 255) can be used as subnets. The subnets are numbered 0 through 255. The following table lists these subnets and their binary and decimal equivalents.

Table 118

| Subnet # | Binary | Decimal |
|----------|----------|---------|
| 0 | 00000000 | 0 |
| 1 | 00000001 | 1 |
| 2 | 00000010 | 2 |
| - | -- | - |
| 254 | 11111110 | 254 |
| 255 | 11111111 | 255 |

- 3 Select the eight most significant bits of the host portion of the Internet address to define the subnets.

These bits are displayed in bold text:

128.001.000.000 = 10000000.00000001.**00000000**.00000000

- 4 Define a subnet mask so that all bits of the network and future subnet fields are set to 1, and all bits of the future host field are set to 0.

Network #: 10000000.00000001.00000000.00000000 = 128.001.000.000

Subnet Mask: 11111111.11111111.11111111.00000000 = 255.255.255.000

This subnet mask (255.255.255.000) must be configured on each host and defined for each router. Use the same subnet mask for devices on the same subnetted subnet that share the same Internet address.

Determine the subnet address for each host.

The 256 subnets have the following addresses:

Subnet #0: 10000000.00000001.00000000.00000000 = 128.001.000.000

Subnet #1: 10000000.00000001.00000001.00000000 = 128.001.001.000

Subnet #2: 10000000.00000001.00000010.00000000 = 128.001.002.000

Subnet #254: 10000000.00000001.11111110.00000000 = 128.001.254.000

Subnet #255: 10000000.00000001.11111111.00000000 = 128.001.255.000

The range of addresses that you can assign for subnet #1 are as follows:

Subnet #1: 10000000.00000001.00000001.00000000 = 128.001.001.000

Low Address: 10000000.00000001.00000001.00000001 = 128.001.001.001

High Address: 10000000.00000001.00000001.11111110 = 128.001.001.254

The range of addresses that you can assign for subnet #35 are as follows:

Subnet #35: 10000000.00000001.00000001.00000000 = 128.001.035.000

Low Address: 10000000.00000001.00000001.00000001 = 128.001.035.001

High Address: 10000000.00000001.00000001.11111110 = 128.001.035.254

The range of addresses that you can assign for subnet #129 are as follows:

Subnet #129: 10000000.00000001.10000001.00000000 = 128.001.129.000

Low Address: 10000000.00000001.10000001.00000001 = 128.001.129.001

High Address: 10000000.00000001.10000001.11111110 = 128.001.129.254

The range of addresses that you can assign for subnet #255 are as follows:

Subnet #255: 10000000.00000001.11111111.00000000 = 128.001.255.000

Low Address: 10000000.00000001.11111111.00000001 = 128.001.255.001

High Address: 10000000.00000001.11111111.11111110 = 128.001.255.254

- 5 Assign the Internet address to the bridge/router.

For example, if subnet #1 is connected to bridge/router port #1, you can enter the following command to assign the Internet address:

```
SETDefault !1 -IP NETaddr = 128.001.001.001 255.255.255.000
```

Subnets: Example 2 The InterNIC assigns you a Class B Internet address of 128.001.000.000. You need to establish four subnets with each subnet capable of supporting up to 16,381 hosts.

To solve this problem, follow these steps:

- 1 Convert the address assigned by the InterNIC to binary format:

For example:

128.001.000.000 = 10000000.00000001.00000000.00000000

The underlined binary digits represent the network portion of the Internet address assigned by InterNIC.

- 2 Determine the number of binary digits you need to represent four subnets.

Two binary digits are required to define 4 subnets ($2^2 = 4$). The binary values of all zeros (decimal value 0) and all ones (decimal value 255) can be used as subnets. For example, the four subnets you select to use can be numbered 0 through 3 and you can select the two most significant bits of the host portion of the Internet address. The following table shows subnet numbers 0 through 3 and their binary and decimal equivalents. The two most significant bits selected are shown in bold.

Table 119

| Subnet # | Binary | Decimal |
|----------|-----------------|---------|
| 0 | 00 00000 | 000 |
| 1 | 01 00000 | 064 |
| 2 | 10 00000 | 128 |
| 3 | 11 00000 | 192 |

- 3 Select the two most significant bits of the host portion of the Internet address to define the subnets.

These bits are displayed in bold text:

128.001.000.000 = 10000000.00000001.**00**000000.00000000

- 4 Define a subnet mask so that all bits of the network and future subnet fields are set to 1, and all bits of the future host are set to 0.

Network #: 10000000.00000001.**00**000000.00000000 = 128.001.000.000

Subnet Mask: 11111111.11111111.**11**000000.00000000 = 255.255.192.000

This subnet mask (255.255.192.000) must be configured on each host and defined for each router. Use the same subnet mask for devices on the same subnetted subnet that share the same Internet address.

- 5 Determine the subnet address for each host.

The four subnets have the following addresses:

Subnet #0: 10000000.00000001.**00**000000.00000000 = 128.001.000.000

Subnet #1: 10000000.00000001.**01**000000.00000000 = 128.001.064.000

Subnet #2: 10000000.00000001.**10**000000.00000000 = 128.001.128.000

Subnet #3: 10000000.00000001.**11**000000.00000000 = 128.001.192.000

The range of addresses that you can assign for subnet #0 are as follows:

Subnet #0: 10000000.00000001.**00**000000.00000000 = 128.001.000.000

Low Address: 10000000.00000001.**00**000000.00000001 = 128.001.000.001

High Address: 10000000.00000001.**00**111111.11111110 = 128.001.063.254

The range of addresses that you can assign for subnet #1 are as follows:

Subnet #1: 10000000.00000001.**01**000000.00000000 = 128.001.064.000

Low Address: 10000000.00000001.**01**000000.00000001 = 128.001.064.001

High Address: 10000000.00000001.**01**111111.11111110 = 128.001.127.254

The range of addresses that you can assign for subnet #2 are as follows:

Subnet #2: 10000000.00000001.**10**000000.00000000 = 128.001.128.000

Low Address: 10000000.00000001.**10**000000.00000001 = 128.001.128.001

High Address: 10000000.00000001.**10**111111.11111110 = 128.001.191.254

The range of addresses that you can assign for subnet #3 are as follows:

Subnet #3: 10000000.00000001.**11**000000.00000000 = 128.001.192.000

Low Address: 10000000.00000001.**11**000000.00000001 = 128.001.192.001

High Address: 10000000.00000001.**11**111111.11111110 = 128.001.255.254

6 Assign the Internet address to the bridge/router.

For example, if subnet #1 is connected to bridge/router port #2, you can enter the following command to assign the Internet address:

```
SETDefault !2 -IP NETaddr = 128.001.064.001 255.255.192.000
```

Subnets: Example 3

The InterNIC assigns you a Class B Internet address of 128.001.000.000. You need to establish 8 subnets with each subnet capable of supporting up to 8,190 hosts.

To solve this problem, follow these steps:

1 Convert the address assigned by the InterNIC to binary format:

For example:

128.001.000.000 = 10000000.00000001.00000000.00000000

The underlined binary digits represent the network portion of the Internet address assigned by InterNIC.

2 Determine the number of binary digits you need to represent six subnets.

Three binary digits are required to define 8 subnets ($2^3 = 8$). The binary values of all zeros (decimal value 0) and all ones (decimal value 255) can be used as subnets. For example, the 8 subnets can be numbered 0 through 7 and you can select the three most significant bits of the host portion of the Internet address to define the subnets. The following table lists the subnets 0 through 7 and their binary and decimal equivalents. The three most significant bits are shown in bold.

Table 120

| Subnet # | Binary | Decimal |
|----------|------------------|---------|
| 0 | 000 00000 | 000 |
| 1 | 001 00000 | 032 |
| 2 | 010 00000 | 064 |

Table 120

| Subnet # | Binary | Decimal |
|----------|------------------|---------|
| 3 | 011 00000 | 096 |
| 4 | 100 00000 | 128 |
| 5 | 101 00000 | 160 |
| 6 | 110 00000 | 192 |
| 7 | 111 00000 | 224 |

- 3 Select the three most significant bits of the host portion of the Internet address to define the subnets.

These bits are displayed in bold text:

128.001.000.000 = 10000000.00000001.**000**00000.00000000

- 4 Define a subnet mask so that all bits of the network and future subnet fields are set to 1, and all bits of the future host are set to 0.

Network #: 10000000.00000001.**000**00000.00000000 = 128.001.000.000

Subnet Mask: 11111111.11111111.**111**00000.00000000 = 255.255.224.000

This subnet mask (255.255.224.000) must be configured on each host and defined for each router. You should use the same subnet mask for devices on the same subnetted subnet that share the same Internet address.

- 5 Determine the subnet address for each host.

The eight subnets have the following addresses:

Subnet #0: 10000000.00000001.**000**00000.00000000 = 128.001.000.000

Subnet #1: 10000000.00000001.**001**00001.00000000 = 128.001.032.000

Subnet #2: 10000000.00000001.**010**00010.00000000 = 128.001.064.000

Subnet #3: 10000000.00000001.**011**11110.00000000 = 128.001.096.000

Subnet #4: 10000000.00000001.**100**11111.00000000 = 128.001.128.000

Subnet #5: 10000000.00000001.**101**11111.00000000 = 128.001.160.000

Subnet #6: 10000000.00000001.**110**11111.00000000 = 128.001.192.000

Subnet #7: 10000000.00000001.**111**11111.00000000 = 128.001.224.000

The range of addresses that you can assign for subnet #3 are as follows:

Subnet #3: 10000000.00000001.**011**00001.00000000 = 128.001.096.000

Low Address: 10000000.00000001.**011**00001.00000001 = 128.001.096.001

High Address: 10000000.00000001.**011**00001.11111110 = 128.001.127.254

The range of addresses that you can assign for subnet #5 are as follows:

Subnet #5: 10000000.00000001.10100001.00000000 = 128.001.160.000

Low Address: 10000000.00000001.10100001.00000001 = 128.001.160.001

High Address: 10000000.00000001.10100001.11111110 = 128.001.191.254

6 Assign the Internet address to the bridge/router.

For example, if subnet #3 is connected to bridge/router port #1, you can enter the following command to assign the Internet address:

```
SETDefault !1 -IP NETaddr = 128.001.096.001 255.255.224.000
```

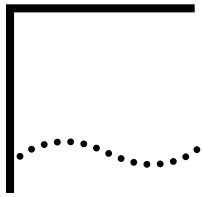
Variable Length Subnet Masks

The 3Com bridge/router supports variable length subnet masks; more than one subnet mask (of different lengths) can be configured for a given network address. In this case, the bridge/router supports up to five different subnet masks for a given network address.

For example, on a Class B network, the following four subnet masks can be assigned for Network 128.1.0.0:

- 255.255.240.0
FF.FF.F0.00 supports 128.1.16.0 – 128.1.224.0
- 255.255.255.0
FF.FF.FF.00 supports 128.1.1.0 – 128.1.15.0
- 255.255.255.252
FF.FF.FF.FC supports 128.1.0.4 – 128.1.0.12
- 255.255.255.240
FF.FF.FF.F0 supports 128.1.0.16 – 128.1.0.240

The subnet masks are stored in the routing tables so that the longest subnet mask takes precedence over the shortest subnet mask.



NSAP AND PSAP ADDRESSING

This appendix provides information about network service access point (NSAP) and presentation service access point (PSAP) addressing, which applies when you are using the bridge/router for Open Systems Interconnection (OSI) routing. It provides the information necessary to establish NSAP and PSAP address values.

NSAP Address Structure

For computer equipment to communicate in a multivendor environment, each device must have a unique address. To accomplish this, a specific portion of every OSI address represents a globally unique location in the open systems environment. This part of the OSI address is the NSAP.

The parts of an NSAP address identify the individual subdomains within the global addressing domain. The NSAP address consists of two parts: the initial domain part (IDP) and the domain specific part (DSP). The IDP contains the authority and format identifier (AFI) and initial domain identifier (IDI).

Table 121 shows the relationship between an NSAP address, IDP, and the DSP addressing scheme for the Government Open Systems Interconnection Profile (GOSIP) version II. The NSAP address contains a maximum of 20 octets or 40 decimal digits.

Table 121 NSAP Address Structure

| IDP | | DSP | | | | | | |
|---------|----------|---------|---------------|----------|----------------|----------|----------|---------|
| 47 | 0005 | DFI | Admin.Author. | Resrvd. | Routing Domain | Area | System | N-SEL |
| 1 octet | 2 octets | 1 octet | 3 octets | 2 octets | 2 octets | 2 octets | 6 octets | 1 octet |

The AFI field specifies the following information:

- Addressing authority responsible for assigning values to the IDI
- IDI format (X.121, E.163, etc.)
- Whether the DSP is in binary or decimal format

Table 122 later in this appendix lists the AFI values according to the IDI formats.

Following the AFI field is the IDI field, which identifies the network addressing authority responsible for determining the format of the DSP. For example, the GOSIP version II specification defines the DSP as containing the following information:

- Data format identifier
- Administration authority
- Reserved
- Routing domain

- Area ID
- System ID
- N-selector

NSAP Address Assignment

Different organizations may have different DSP structures and values. The U.S. government has specified its use of NSAP addresses in GOSIP; your organization can specify its own use of NSAP addresses. However, it is the network administrator's responsibility to determine the proper means for obtaining globally unique addresses.

When you assign an NSAP address to your server, follow these rules:

- The IDI is always in decimal.
- The DSP can be in decimal or hexadecimal. If the DSP is in hexadecimal, it must contain an even number of digits.
- A station ID can be either a logical ID or the unique physical address associated with some communications medium. (This type of physical address is called a Subnetwork Point of Attachment, or SNPA.)
- The preceding slash (/) in the NSAP address is mandatory. Slashes may be used as optional separators before the IDI and DSP (0005 and 01ABCDEF, respectively, in the example below). However, if one of these optional slashes is used, the other must be present also.

Example In this example, GOSIP version II specifies the AFI to be 47, indicating that the IDI value comes from ISO 6523-ICD (International Code Designator). The U.S. GOSIP program has been assigned International Code Designator 0005 by the British Standards Institute (BSI). The Government Services Administration (GSA) can administer the values of the DSP for various U.S. federal offices.

For instance, the GSA can assign an administration authority ID to the Department of Agriculture. The Department of Agriculture can then assign routing domains to its branches (for example, different subnet IDs to branch offices in different states). Each branch office can administer the area and system IDs of its equipment.

If the AFI is an odd number, the DSP is in binary; an even AFI indicates that the DSP is in decimal.

Figure 461 shows an example of the NSAP address using the mandatory preceding slash only.

/47000501ABCDEF000000010003080002000ACE01

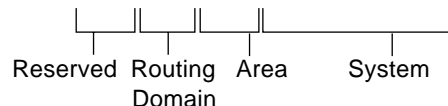


Figure 461 NSAP Address with Mandatory Preceding Slash

The following is an example of an NSAP address using the mandatory preceding slash as well as the optional slashes:

```
/47/0005/01ABCDEF00000001000308000200DACE01
```

Default NSAP Values

3Com bridge/routers are shipped with default values for the AFI, IDI, and prefix DSP fields. Values for the ID and selector fields are generated at boot time.

The following is an example of an NSAP address assigned by 3Com:

```
/49/0053080002A0089D00
```

Table 122 shows the values of individual fields of this NSAP address.

Table 122 Example of an NSAP Address Assigned by 3Com

| IDP | | DSP | | | |
|-----|------|-----------------|-----------|--------------|-------|
| AFI | IDI | Organization ID | Subnet ID | MAC address | N-SEL |
| 49 | Null | 00 | 53 | 080002A0089D | 00 |

For explanations of each of these fields, see “NSAP and PSAP Address Field Definitions” later in this chapter.

Values Derived from NSAP Addresses

From an NSAP address, the following values are derived:

- Area address
- Network Entity Title (NET)

The area address is the NSAP address without the ID and selector fields. It consists of the AFI, the IDI, and the prefix of the DSP. The area address is critical to intermediate system-to-intermediate system (IS-IS) routing operations.

NET is an NSAP address with the selector value of 0. NET is used for IS-IS and Connectionless Network Protocol (CLNP) operations.

NSAP Registration Authorities

In the U.S., there are two registration authorities: the American National Standards Institute (ANSI) and the Government Services Administration (GSA). The GSA is the registration authority for all NSAP addresses that follow the U.S GOSIP version II NSAP address format. The NSAP address format is as follows:

```
AFI =      47 (1 octet)
IDI =      0005 (2 octets)
DSP =      Version
           DFI
           AAI
           Routing domain
           Area ID
           System ID (6
           octets)
           Selector (1 octet)
```

You can obtain registration information by writing to the following authority:

Government Services Administration
Office of Telecommunication Services
Registration Services, Room 1221-L KBA
18th and F Street N.W.
Washington D.C. 20405

PSAP Addresses

The PSAP address contains the NSAP address and a full set of (N)-Selector: T-selector, S-selector, and P-selector.



PSAP addresses are used on the bridge/router for OSI connection services only. For normal OSI routing, use the NSAP addressing scheme.

There are three layers above the Network Layer that require addressing information: Transport, Session, and Presentation. In OSI terminology, these addresses are called (N)-selectors, where N is an OSI layer. The corresponding selectors are termed T-, S-, and P-selectors. In an open system, the combination of these selectors uniquely identifies an application entity.

The NIST Implementation Agreements specify that maximum lengths of 32, 16, and 4 octets for the T-, S-, and P-selectors be supported. Selectors for open systems from different vendors may differ in length or value.

On the 3Com OSI connection service, the P-selector portion of the PSAP address is used by the Virtual Terminal Protocol (VTP) to map to an X.25 address. The following shows the syntax for the complete PSAP address:

```
<NSAP address> | <T-SEL> | <S-SEL> | <P-SEL>
```

(N)-selectors provide the local addressing elements for accessing OSI-layer protocol processes on the server or host equipment at the destination address. In the 3Com syntax of the OSI address, (N)-selector values follow the NSAP address. Each (N)-selector field is preceded by the (|) character.

On 3Com servers, the (N)-selector values are the names of the protocol module processes operating in the respective OSI model layers. The special character "!" may be used to represent the T-selector and S-selector fields in a 3Com PSAP address. For example, the address can be in the following syntax:

```
<NSAP address>!<P-SEL>
```

The exclamation mark (!) in the syntax is interpreted as the T-selector and S-selector values. For the actual values of these selectors in a 3Com address, see "NSAP and PSAP Address Field Definitions" later in this chapter.

You cannot assume that the (N)-selector values of the destination server or host are the same as those on the local system. Other environments may choose to omit the use of (N)-selectors or use simple numeric values. If the destination server or host does not have or need a full set of (N)-selectors, the absence of an (N)-selector must be indicated with an empty field.

The following example specifies a PSAP address with absent transport (T-SEL) and session (S-SEL) layer selector values:

```
<NSAP address> | | | <P-SEL>
```

NSAP and PSAP Address Field Definitions

This section describes each field in the NSAP and/or PSAP address relevant to bridge/router operation:

| | |
|-------|--|
| AFI | The Authority and Format Identifier contains two decimal digits. In 3Com syntax, this field is always preceded with a slash (/), which identifies the NSAP portion of an NSAP or PSAP address. The AFI specifies the official body responsible for allocating IDI field values, the format of the IDI field, and whether the syntax of the DSP should be specified with binary or decimal digits. |
| IDI | The Initial Domain Identifier contains up to 15 decimal digits depending on IDI format established in the AFI field. In 3Com syntax, this field may be preceded with a slash(/). It identifies the network addressing authority responsible for determining the format of the DSP field. The IDI field always follows the AFI field. |
| DSP | The format and length of the Domain Specific Part is determined by the combined AFI and IDI fields. In 3Com syntax, this field may be preceded with a slash (/). In one case, the DSP field may contain the organization ID, network number or subnet ID, and MAC address fields to provide additional levels of addressing for networks such as those described in the GOSIP specification. In another case, it may contain an Internet or Ethernet address for local OSI networks. |
| N-SEL | The length of the N-selector field is always a single octet. On 3Com servers or bridge/routers, the value of this field is a one. 3Com servers and bridge/routers use this field to identify the client of the Network layer of the OSI model, which is always the OSI Transport protocol. A special value of 0 is used to identify the network entity itself, which forms the Network Entity Title (NET). |
| T-SEL | The T-selector field contains up to 32 octets according to the NIST agreements. The T-selector values are vendor-dependent. 3Com servers use "S" and "E" as the first and second octets in this field, respectively, to identify the ISO Session Protocol. 3Com servers use this field to identify the client of the Transport layer of the OSI model. |
| S-SEL | The S-selector field contains up to 16 octets according to the NIST agreements. 3Com servers use "P," "R," and "E" as the first, second, and third octets in this field, respectively, to identify the ISO Presentation Protocol. 3Com servers use this field to identify the client of the Session Layer of the OSI model. |
| P-SEL | The P-selector field in the Presentation Address is 2 octets in length for the 3Com OSI Connection Service and the value of the first octet must be either 0 or 4. When you want to make a connection using Telnet profiles, 0 is used, and 4 is used for the X.3 profiles. As a result, the mapping is only for the second octet of the P-selector. |

Table 123 lists the AFI values and their associated IDI formats, as described in Addendum 2 of the International Standard 8348. The AFI value identifies the abstract syntax (decimal or binary format) for the DSP portion of the NSAP address. The IDI formats identify the addressing authority responsible for assigning values of the DSP.

Table 123 AFI Values

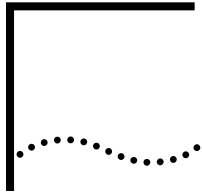
| IDI Format | AFI Values | |
|------------|-------------------------------|------------------------------|
| | Decimal
(Max. DSP Length*) | Binary
(Max. DSP Length*) |
| X.121 | 36 (24) | 37 (9) 12 [†] |
| ISO DCC | 38 (35) | 39 (14) 17 [†] |
| F.69 | 40 (30) | 41 (12) 15 [†] |
| E.163 | 42 (26) | 43 (10) 13 [†] |
| E.164 | 44 (23) | 45 (9) 12 [†] |
| ISO ICD | 46 (34) | 47 (13) 17 [†] |
| Local | 48 (38) | 49 (15) 19 [†] |

* Decimal digits for decimal; binary octets for binary.

† Maximum length of binary DSP reflect change in Standard 8348 pDAM3.

Each of the IDI format values is described here:

- X.121 The IDI format adheres to CCITT Recommendation X.121. The maximum IDP length is 16 digits.
- ISO DCC The IDI format adheres to values allocated by an ISO DCC IDI Format Registration Authority. The IDP length is 5 digits.
- F.69 The IDI format adheres to CCITT Recommendation F.69. The maximum IDP length is 10 digits.
- E.163 The IDI format adheres to CCITT Recommendation E.163. The maximum IDP length is 14 digits.
- E.164 The IDI format adheres to CCITT Recommendation E.164. The maximum IDP length is 17 digits.
- ISO ICD The IDI format adheres to International Code Designator (ICD) values allocated by ISO 6523. The IDP length is 6 digits.
- Local The IDI is a null value. The local network administrator is responsible for allocating the values of the DSP. The IDP length is 2 digits.



SUPPORTED MIBs

This appendix lists all management information base (MIB) modules supported by the NETBuilder family of products and the software packages that run on NETBuilder systems. To determine which MIB modules are supported by a particular software package, obtain the list of object IDs assigned to 3Com products by anonymous ftp from ftp.3com.com. The file 3com-products.mib in the directory pub/3com-mibs/all-mibs/ contains this list.

Supported Operations

The Get and GetNext operations are supported for all simple objects and tables. The Set operation is supported with limitations on all objects that provide write access. Set operations take effect immediately, and changes are saved to the disk so new configurations are not lost after reboot.

The tables within the 3Com-defined MIBs support a subset of the functionality provided by the rowStatus textual convention. Row creation is allowed using only the *createAndGo* method. With the *createAndGo* method, the *Status* object of the table is set to *creatAndGo(4)* within the same protocol data unit (PDU) that carries the other columnar values; the result is that the new row is immediately marked as *active(1)*. Once active, the row cannot be modified (there are a few exceptions). Changes to rows can be made only by first deleting the row, and then recreating it with the proper values.

Port Numbering Convention in SNMP

Throughout this appendix, references to port-numbering assume the format used in the NETBuilder II user interface. The current implementation of Simple Network Management Protocol (SNMP) uses a port-numbering scheme that differs from the NETBuilder II user interface. Table 124 shows the relationship between these two schemes.

Table 124 Port Numbering in SNMP

| UI Port Label | 8-Slot NB II
SNMP Port Label | UI Port Label | 8-Slot NB II
SNMP Port Label |
|---------------|---------------------------------|---------------|---------------------------------|
| 1/1A | 1 | 5/5A | 5 |
| 1B | 9 | 5B | 13 |
| 1C | 17 | 5C | 21 |
| 1D | 25 | 5D | 29 |
| 1E | 33 | 5E | 37 |
| 1F | 41 | 5F | 45 |
| 1G | 49 | 5G | 53 |
| 1H | 57 | 5H | 61 |
| 2/2A | 2 | 6/6A | 6 |
| 2B | 10 | 6B | 14 |

Table 124 Port Numbering in SNMP (continued)

| UI Port Label | 8-Slot NB II
SNMP Port Label | UI Port Label | 8-Slot NB II
SNMP Port Label |
|---------------|---------------------------------|---------------|---------------------------------|
| 2C | 18 | 6C | 22 |
| 2D | 26 | 6D | 30 |
| 2E | 34 | 6E | 38 |
| 2F | 42 | 6F | 46 |
| 2G | 50 | 6G | 54 |
| 2H | 58 | 6H | 62 |
| 3/3A | 3 | 7/7A | 7 |
| 3B | 11 | 7B | 15 |
| 3C | 19 | 7C | 23 |
| 3D | 27 | 7D | 31 |
| 3E | 35 | 7E | 35 |
| 3F | 43 | 7F | 47 |
| 3G | 51 | 7G | 55 |
| 3H | 59 | 7H | 63 |
| 4/4A | 4 | 8/8A | 8 |
| 4B | 12 | 8B | 16 |
| 4C | 20 | 8C | 24 |
| 4D | 28 | 8D | 32 |
| 4E | 36 | 8E | 40 |
| 4F | 44 | 8F | 48 |
| 4G | | 8G | 56 |
| 4H | | 8H | 64 |

**MIBs Supported by
the Bridge/Router**

The bridge/router supports the following SNMP MIB modules defined by the Internetworking Engineering Task Force (IETF), MIB modules are defined by 3Com, IBM, and Novell.

The IETF MIB modules are:

- RFC 1213 (MIB II with interface group obsoleted by RFC 1573)
- RFC 1243 (AppleTalk MIB)
- RFC 1286 (Bridge MIB)
- RFC 1284 (Ethernet-like MIB)
- RFC 1285 (FDDI MIB)
- RFC 1315 (Frame Relay DTE MIB)
 - Except for frCircuitCommittedBurst and frCircuitExcessBurst
- RFC 1354 (IP Forwarding MIB)
- RFC 1573 (MIB II ifTable, ifStackTable, ifxTable)
- RFC 1253 (OSPF MIB)
- RFC 1271 (RMON Alarm and Event MIB)
- RFC 1659 (RS232 Hardware Devices)
- RFC 1593 (APPN MIB)

- RFC 1749 (Source Route MIB)
- RFC 1231 (Token Ring MIB)
- RFC 1304 (SMDS Interface Protocol (SIP) MIB)
- RFC 2127 (ISDN MIB using SMIv2)



The Token Ring MIB was moved from underneath the experimental branch to the transmission branch per RFC 1239. The 3Com implementation of the Token Ring MIB supports the dot5Table and the dot5StatsTable, but it does not support the optional dot5Timer Table.



The LAN emulation client (LEC) MIB is used in conjunction with the MIB II "ifTable" objects to allow SNMP network management of all aspects of an Emulated LAN. The complete interpretations of the "ifTable" to LEC MIB objects mapping is described in Section 9.0 of the LAN Emulation Client Management Specification, Version 1.0, ATM Forum Technical Committee

The 3Com private MIB modules control the following bridge/router services:

- AuditLog
- Bridge extension
- DLSw
- DLSw Topology
- DVMRP
- IP
- IP RIP
- IP security options
- IPX
- IPX policies
- LLC
- Multicast IP
- Multiple logical networks
- Mnemonic filtering
- Performance
- Port and path
- PnP VPN
- Remote Access Server
- Router Discovery Protocol
- SDLC
- SHDLC
- System



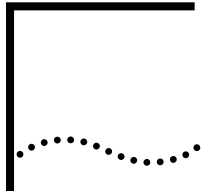
The IBM-defined MIB controls the DLSw Service. The Novell-defined MIB controls the NLSP Service.

3Com Private MIBs

The 22 3Com private MIBs are located under the following headings on the MIB disk:

| | |
|---------------------------|---------------------------|
| ■ AuditLog MIB | A3Com-AUDL-MIB |
| ■ Bridge Extension MIB | A3Com-Bridge-MIB |
| ■ LLC MIB | A3Com-LLC-MIB |
| ■ DLSw MIB | A3Com-DLSw-MIB |
| ■ DLSw Topology MIB | A3Com-DLSw-Topo-MIB |
| ■ MLN MIB | A3Com-MLN-MIB |
| ■ Filtering MIB | A3Com-Filter-MIB |
| ■ IP Extension MIB | A3Com-IPextns-MIB |
| ■ IP Security Options MIB | A3Com-IPSO-MIB |
| ■ IPX MIB | A3Com-IPX-MIB |
| ■ IPX Policies MIB | A3Com-IPXpolicy-MIB |
| ■ Port and Path MIB | A3Com-PortPath-MIB |
| ■ RIP IP MIB | A3Com-RIP-IPextns-MIB |
| ■ SDLC MIB | A3Com-SDLC-MIB |
| ■ System MIB | A3Com-System-MIB |
| ■ Multicast IP MIB | A3Com-Mip-MIB |
| ■ DVMRP MIB | A3Com-Dvmrp-MIB |
| ■ PerformanceMIB | A3Com-Perf-MIB |
| ■ PnPVPn | A3Com-PnPVPn-MIB |
| ■ Remote Access | A3Com-NB-RemoteAccess-MIB |
| ■ Router Discovery | A3Com-Rdp-MIB |
| ■ SHDLC | A3Com-Shdlc-MIB |

To get a listing of the levels of MIB support offered by each bridge/router product, ftp to ftp.3Com.com, enter the log-on command as anonymous, enter the cd command to change the directory to pub/docs/3Com-mibs, and enter the get command to obtain the README file.



MACRO FEATURES

This appendix provides information about macro conventions and macros with conditional statements.

Macro Conventions

Macro contents must begin with a left parenthesis. If the definition requires more than one line, press the Return key after the opening left parenthesis. The macro prompt then appears as a locator for you. All characters entered between the opening and closing parentheses are part of the macro. Nested parentheses in balanced pairs are allowed. When you end the macro with the closing right parenthesis, the normal server prompt returns.

A single macro cannot contain more than 256 characters. Macro names must follow the DOS file naming conventions: the macro name cannot contain more than 14 characters and the macro name extension, if any, cannot contain more than three characters.

If an error is detected in the macro, the macro stops executing, and an error message appears. The error message includes the macro name and a short explanation. After the message appears, you are returned to Command Mode.

A macro can include the DO command to call another macro. Embedded calls to other macros is called *nesting*. Because of the large amount of memory required to keep track of the calling history and variables, the limit for nested macros is 10.



If you use command substitution in macros, the user interface may hang because of the echoing of flow-control characters between the quotation marks in the string sent from the bridge/router.

Macros With Conditional Statements

Macros with conditional statements contain variables (such as arguments and return codes) and control structures (such as "if-else-end" and "switch-case-end"). Control structures instruct the macro to test conditions or make comparisons from which execution decisions can be made. Variables contain the values from which these comparisons are made. The execution decisions affect the final macro output.

Macro Variables

Macro variables store values in memory, which can be evaluated by a macro during execution. Values can be either numeric or strings, and are represented by a variable name such as \$1 or \$rc. Examples of variable types are arguments, input or output requests, return codes, or global variables. A variable can be readable, writable, or both readable and writable by the macro.

Table 125 lists all possible variables within macros. The table breaks down the variables by type (argument, input, output, return code, asynchronous event, global, system/user), value (numeric or string), and actions (readable or writable). Each variable name begins with a dollar sign (\$).

Table 125 Variables within Macros

| Type | Name | Value
Numeric (N)/ String (S) | Readable | Writable |
|----------|---------|----------------------------------|----------|----------|
| Argument | # | N | * | * |
| | 1 | S | * | * |
| | 2 | S | * | * |
| | 3 | S | * | * |
| | 4 | S | * | * |
| Input | 5 | S | * | * |
| | < | S | * | |
| | > | S | | * |
| | rc | N | * | |
| | error | S | * | * |
| Event | brk | S | * | * |
| | global | N | * | * |
| Global | lpw | S | * | |
| | sess | N | * | |
| | prompt | S | * | |
| | portid | N | * | |
| | priv | N | * | |
| | user | N | * | |
| | nm | N | * | |
| | lnm | N | * | |
| | gnm | N | * | |
| | eth_add | S | * | |
| lcerrs | N | * | | |

Variable Types

There are nine types of variables:

- Argument (local)
- Input
- Output
- Return code
- Asynchronous
- Global
- System/user information
- Numeric and String
- Readable and Writable

These variables are described in the following sections.

Argument (local) Argument (local) variables (\$#, \$1, \$2, \$3, \$4, and \$5) provide the option of passing up to five arguments to a macro. Within the macro, these arguments can be referenced by \$1 through \$5. The \$# variable contains the actual number of arguments passed. Local argument variables apply only to a particular macro. For example:

```

do <macro-name>      $1      $2      $3      $4      $5 (empty)
                    call    me      at      370-6610

```

The \$1 variable will contain call, \$2 will contain me, \$3 will contain at, \$4 will contain 370-6610, and \$5 is empty. The \$# variable will be 4.

Similar to a C language procedure, argument variable values exist only within the macro, and these values disappear when the macro terminates. The same macro can be executed with different argument variables assigned, giving it a completely independent value. For nested macros, the called macro has its own set of argument variables, independent of the calling macro.

Input Input variables (\$<) cause the macro to stop executing and wait for your input. \$< is then substituted by your input, and the macro continues executing based on your input. Input variables are illegal in macros that are submitted to the SCHeduling Service.

Output Output variables (\$>) cause any string of characters assigned to this variable to be displayed on your terminal. Output variables can generate all 127 characters on the terminal screen, as does the Echo command. For example, both of the following lines generate a bell ([Ctrl]+G) to the terminal when they are executed:

```

echo "^G"
$> = ^G

```

However, \$> does not operate exactly the same as the Echo command. The Echo command automatically appends a CR-LF after the string being echoed, and the \$> variable does not. Therefore, \$> is more convenient to use for controlling screen layout.

Return Code Return code variables (\$rc) contain the return status of the last executed user interface command. \$rc is always 0 (no errors) if the last command executed successfully. When a called macro returns to the calling macro, the \$rc variable is not affected by the return operation.

Asynchronous Event Asynchronous event variables (\$error and \$brk) handle unexpected conditions that cause a macro to abort. A macro will abort under one of two conditions: when an internal error is detected or when the user presses the Break key.

The \$error variable is used to recover from an error. With \$error defined, if an error occurs, the macro will stop executing, and a new macro, as specified by the \$error variable, will automatically begin executing to clean up or recover from the error. Without \$error defined, an error detected in a macro stops the execution of the macro and returns to command mode.

The \$break variable defines a macro that will begin executing when the user presses the Break key. For example, you can define a macro called "recover" that will put the user into Listen mode, then assign the macro "recover" to \$break so that recover is executed when the Break key is pressed.

Without the \$break variable defined, you can exit a macro while it is executing by pressing the Break key (unless the NoMacroBreak option is set in the InterAction parameter). The Break key exits the macro and returns to Command mode.

The \$error and \$brk variables do not cancel the effects of errors or breaks. They restart a new macro service in order to handle the error or break signal. If you do not define \$error or \$brk, errors and break signals will force the macro to stop executing, and will return to command mode.

In addition to a macro name, \$error and \$brk can contain up to five arguments. For example:

```
$error = <macro name for handling error> arg1 arg2 ...
```

For descriptions of these variables, see "Argument (local)" earlier in this chapter.

Global Global variables (\$global) provide another way to pass information between macros when calling a macro. A global variable is a variable that is globally shared among all macros executed from the same user port.

You can use \$global to test the return status of a macro.

System/user Information System/user information variables include the following:

| | |
|-----------|--|
| \$eth_add | Server Ethernet address - media access control (MAC) address of the first interface |
| \$lpw | Local password |
| \$portid | User port number executing the macro |
| \$prompt | Prompt strings, depending on user privilege |
| \$sess | Number of user sessions outstanding (on this port) |
| \$priv | Privilege of the user executing the macro |
| \$user | User privilege value is 0 (user) |
| \$lnm | User privilege value is 1 |
| \$gnm | User privilege value is 1 |
| \$nm | Network manager, 1 |
| \$lcerrs | Number of failed commands since the last execution of LoadConfigs. Value of -1 if no LoadConfigs command has executed. |

\$priv contains the current privilege level of the user. Its value will be equal to \$user or \$nm depending on the privilege level of the user. \$priv tests the privilege level of the user within a macro. \$lpw contains the passwords for Network Manager privilege levels. These variables are used to compare passwords within a macro.

Numeric Variables and String Numeric variables store decimal values between 32767 and -32768. String variables can store any numeric value (within the described limit) or any character sequence. String variables can perform all the functions of numeric variables. They can be compared with other strings or

numeric values, incremented or decremented, or assigned to another numeric variable. Numeric values incremented beyond 32767 become negative.

Readable and Writable All variables, except the output variable \$>, are readable by the macro, which means that the values they store can be interpreted and compared by the macro in any expression. The \$> variable generates output to the user's screen.

Some variables are writable by the macro, which means they can be reassigned new values within the macro. As shown in Table 125, only the following variables are writable:

| | |
|-----|----------|
| \$# | \$5 |
| \$1 | \$> |
| \$2 | \$error |
| \$3 | \$brk |
| \$4 | \$global |

Assigning values to non-writable variables causes a syntax error and aborts the macro. For example, if you want to change the global password, \$lpw = <password> will not work because the variable \$lpw is not writable.

Comparing and Reassigning Variables

Six comparison operators are available for testing macro variables against each other or constant values. Comparison operators are used most often in the if-else-end control structure to compare values. Table 126 lists the available comparison operators.

Table 126 Comparison Operators

| Operator | Comparison Performed |
|----------|---|
| == | Values are equal. |
| != | Values are not equal. |
| >= | Value on left is greater than or equal to value on right. |
| <= | Value on left is less than or equal to value on right. |
| < | Value on left is less than value on right. |
| > | Value on left is greater than value on right. |

Numeric and string variables can be compared with each other. These rules apply:

- When both variables are numeric, which can be string variables containing numeric values, the comparison is based on value. For example:
123 == 00123
- If any one of the variables is a string value (containing a character other than 0–9, excluding space and tab), a string comparison is performed. The difference between uppercase and lowercase is ignored. Only the first character is compared. If the first characters are equal, the next character is compared until a decision is made. For example:
ABC is equal to AbC
ABC is not equal to 123
- Variables can be reassigned with statements such as the following:
<variable> = <value>

- Only numeric values or string variables containing numeric values can be assigned to numeric variables. Otherwise, a syntax error is detected and the macro execution is aborted.
- Numeric variables can be incremented or decremented with plus and minus statements. The plus statement is used most often within a loop structure to increment a counter, which can then be tested against a value.

```
variable ++
```

```
variable --
```

Variable Substitutions

Immediately before a line is executed, the line is scanned and all variables are replaced with their values. Substitution can be done only once. Variables can appear anywhere in the line and still be substituted. For example:

```
Echo "My arguments are $# $1 $2 $3 $4 and $5"
REMOte 192.9.200.$1
```

Two dollar signs (\$) allow you to escape variable substitutions. For example, if you enter:

```
Echo "argument $$1 is $1"
```

the following display appears:

```
argument $1 is <substituted value>
```

Control Structures

Control structures are the tools that can alter the sequences of execution. The syntax is similar to a C program. Control structures must begin and end within the boundary of the macro. For example, in the if-else-end structure, all three parts of the conditional statement (if, else, and end) must be contained within the macro. If any part of the structure is missing, a syntax error is detected and the macro aborts.

Control structures are free to nest within one another. For example, within one loop structure you can have several if-end structures. There is no limit to the number of nested control structures allowed.

If-Else-End

The if-else-end structure is used to make two-way decisions. The syntax is as follows:

```
if <expression>
commands ...
else
commands ...
end
```

The else part is optional. The <expression> is evaluated; if it is TRUE, the macro executes the immediately following commands. If it is FALSE and there is an else statement, then the commands following the else statement are executed. If it is FALSE and there is no else statement, then the commands following the end statement are executed. There can be any number of commands between if-else-end, including none.

The syntax for <expression> is:

```
<variable> <op> <value>
```

<op> can be one of the six comparison operations ==, !=, >=, >, <=, and <. A single variable must be on the left side of the comparison operator. <value> can be any string of characters and digits, with variables intermixed, or it can be empty.

Both <variable> and <value> can contain numeric values or strings. Both can contain more than one word, but only the first word is compared.

Switch-Case-End

The switch structure is a multiway decision maker. It is usually used with \$< to make a comparison based on the user's input. The syntax is as follows:

```
switch <value>
  case <value>
    commands ...
  case <value>
    commands ...
  case *
    commands ...
end
```

The switch structure tests whether the <value> immediately following the switch matches one of the <values> after case. If the values match, the macro executes the immediately following commands.

There can be any number of commands after each case, including none.

In situations where there is no match, but there is a case* (wildcard character) before the end, the command following case* is executed. The * can appear after any case between the switch and end. The case* is not required in the switch-case-end structure.

If the value immediately following the switch does not match any case within the switch, the macro will continue to execute the commands after the end.

Loop-End

Any of the following commands can appear within the loop structure:

```
loop
  commands ...
end
```

The loop structure comprises a set of instructions that can be executed repeatedly while certain conditions prevail. You define the conditions within the loop structure using the comparison operators. The loop can be terminated by using the "break," "return," and "exit" keywords inside the loop-end structure.

Keywords If a keyword is not the first word in a line, it is not recognized. The following keywords can be used:

```
audit          exit
```

| | |
|----------|--------|
| break | if |
| case | loop |
| continue | return |
| else | switch |
| end | |

Audit

This keyword generates an audit trail record of the macro information (MI) type. You can provide a string of data following the keyword.

Break

This keyword terminates the current loop structure, and the macro execution continues after the end keyword. The current loop structure is the structure that contains the break keyword. The break function does not apply to the switch-end structure. Break is only meaningful within a loop-end structure.

Continue

This keyword directs macro execution to the beginning of the enclosing loop-end structure. It is only meaningful within a loop-end structure.

Exit

This keyword stops execution of all macros. It frees all associated buffers and returns the user to Command mode.

Return

This keyword stops execution of the current macro. It resumes the previous calling macro, if any.

Macro Caching and Shared Macros

As macros become larger, more complex, and more heavily nested, even a relatively simple macro can require a large number of nested macros. The dependence on macro file service from the floppy diskette on the local server becomes more and more critical. A sudden failure of the local disk drive can create a serious service interruption. The server has the following features to handle such failures:

- The server keeps track of all the macros currently being executed in a macro cache.

When a macro is not being executed, it is kept in the cache memory as long as there is space available. The next time you request a macro file, the cache is searched first. If the macro is found, it is automatically executed. This reduces the dependence on the server and speeds up macro execution.

- The server links several users to a single copy of a macro instead of distributing many copies of it, thereby sharing the macro.

Sharing macros relieves the memory overhead associated with keeping numerous similar macros. There is no limit to the number of users that can be linked to a macro.

As long as a macro is linked to a user, that macro stays cached. If there are no users linked to the macro, the server keeps the macro in the cache based on the amount of space available in the cache.

For example, if the macro cache is between 80% and 100% full, a macro that is not linked to any users will be stored for up to 10 minutes.

Table 127 lists the macro cache aging algorithm.

Table 127 Macro Cache Aging

| Cache Usage Level | Aging |
|--------------------------------|-------------------------|
| Below 50% | No aging |
| Below 80% but higher than 50% | Cached up to 8 hours |
| Below 100% but higher than 80% | Cached up to 10 minutes |

If the cache overflows, the server rebuilds its cache memory, which frees all macros that are not linked.

Macro caching can cause a discrepancy between the DO <macro name> and SHow MACros <macro name> commands. The DO command searches for the file first in the cache and then in the local diskette or macro server. The SHow command reads the macro file directly from the local diskette or macro file server and never checks the cache. If the file stored in the cache is not the same as the one on the diskette or file server, you get different results.

If the network manager modifies the macro files, the cache aging algorithm may not pick up new macros until after the aging period. The FLush MACros command is available to force the server to flush its cache.

Larger Macros

The size limit for each macro that is stored is 256 bytes. In many applications, this is barely enough for a moderately sophisticated macro. To solve this problem, the network manager can create macros that are the necessary size using the plus sign (+).

To store a macro larger than 256 bytes, the macro must be split into smaller macros such as m1, m2. The name of the large macro will be m1+m2 when it is cached in the memory. The macro cache stores up to eight bytes of the macro name. When the cache is searched for a macro, only the first 14 bytes of the macro name are considered significant. If a macro name exceeds the limit, it is truncated. For example, the following two macro names are considered the same in the cache:

```
macro1+macro2+macro3
macro1+macro2+XX
```

Spaces are not allowed around the plus sign (+). To execute the macro, use the DO command m1+m2, which informs the server which two macros must be read and concatenated into one large macro. The command DO m1<space>+m2<space> means execute +m2 as its argument.

You can break up control structures across two or more macros that will be concatenated with the plus sign (+), because concatenated macros are considered one macro. For example, macro1 could contain the if part of the if-else-end control structure, and macro2 could contain the else and end parts of the structure.

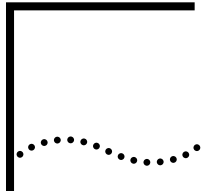
There is no set limit on the number of plus signs (+) that you can use; however, the + operation requires a great deal of memory from the large server buffers and should not be overused.

Macro Nesting A macro can call other macros (including itself), similar to a subroutine call in any computer language. These calls to other macros create macro nesting.

The limit to the number of nested macros is 10. Exceeding this limit causes an error and aborts the macro service. The variable \$error can be set up to automatically capture the error and start a new macro service.

Example The following example shows macros using features such as variables and conditional statements. This example executes the TraceRoute command on the address specified in the first argument to the macro and loops the number of times specified in the second argument to the macro.

```
NETBuilder [1]# define tracen = (  
if $# !=2  
echo "USAGE: tracen <destination -IP-addr> <times-to-loop>"  
exit  
else  
if $2 ==0  
echo "TRACEN: iterations must be > 0"  
exit  
end  
loop  
echo " "  
$>=#$2-  
traceroute $1  
$2--  
if $2 <= 0  
exit  
end  
end  
end)
```



STATISTICS DISPLAYS

This appendix provides displays of accumulated system statistics for a particular service. To display statistics, use the `SHow -SYS STATistics -<service> <option>` syntax.

The statistics displayed are based on the time period in which they have been gathered. For example, during the busiest minute, specified by the `SETDefault -SYS SampleOption` command, and on the time interval in which you want to see the statistics, specified by the `SETDefault -SYS SampleTime` command.

A statistical display showing a blank in any field indicates that the service was not configured for the specified port.

The `FLush -SYS STATistics -service` command may take several seconds before statistics sampling is restarted. For more information on the `FLush` command, see the `SYS Service Parameters` chapter in *Using Enterprise OS Software*.

For more information on syntax, see *Reference for Enterprise OS Software*. For more information on displaying FR and X25 statistics, see the `FR Service Parameters` and `X25 Service Parameters` chapters in *Reference for Enterprise OS Software*.

This appendix provides the statistics displays in alphabetical order.



The displays in this appendix are examples only. Actual displays will vary according to your system configuration.

AppleTalk Service

The following is an example of a display generated by the `SHow -SYS STATistics -AppleTalk` command:

```
ACCUMULATED VALUES
== AppleTalk statistics ===== 1===== 3===== 5===== 7=====

DDP Statistics :
  General Datagram Counts :
    Locally Originated           16          65          -          -
    Short DDP Out                 0           0          -          -
    Long DDP Out                 16          65          -          -
    Total In                     25         104          -          -
    In - Not Local Dest          0           0          -          -
    In - Locally Destined        25         104          -          -

Dropped Datagram Counts :
    No Recipient                 25          55          -          -
    No Route                     0           0          -          -
    Data Too Short               0           0          -          -
    Data Too Long               0           0          -          -
    Broadcast Error              0           0          -          -
```

| | | | | |
|-------------------------|----|----|---|---|
| Short Header Error | 0 | 0 | - | - |
| Hop Count Error | 0 | 0 | - | - |
| Checksum Error | 0 | 0 | - | - |
| RTMP Statistics : | | | | |
| Network Filter Matches | 0 | 0 | - | - |
| Data/Responses In | 0 | 0 | - | - |
| Data/Responses Out | 5 | 5 | - | - |
| Requests In | 0 | 0 | - | - |
| Requests Out | 10 | 10 | - | - |
| Route changes (= dist) | 0 | 0 | - | - |
| Route changes (shorter) | 0 | 0 | - | - |
| Network Dist. Exceeded | 0 | 0 | - | - |
| Network Route Deletes | 0 | 0 | - | - |
| Invalid Packets | 0 | 0 | - | - |
| Bad Tuple Packets | 0 | 0 | - | - |
| Net Number Overlaps | 0 | 0 | - | - |
| ZIP Statistics : | | | | |
| Queries In | 0 | 0 | - | - |
| Queries Out | 0 | 0 | - | - |
| Replies In | 0 | 0 | - | - |
| Replies Out | 0 | 0 | - | - |
| Extended Replies In | 0 | 0 | - | - |
| Extended Replies Out | 0 | 0 | - | - |
| GetZoneList Req. In | 0 | 2 | - | - |
| GetZoneList Rep. Out | 0 | 2 | - | - |
| GetLocalZones Req. In | 0 | 0 | - | - |
| GetLocalZones Rep. Out | 0 | 0 | - | - |
| GetMyZone Req. In | 0 | 0 | - | - |
| GetMyZone Rep. Out | 0 | 0 | - | - |
| GetNetInfo Req. In | 0 | 2 | - | - |
| GetNetInfo Rep. Out | 0 | 2 | - | - |
| GetNetInfo Req. Out | 0 | 0 | - | - |
| GetNetInfo Rep. In | 0 | 0 | - | - |
| Invalid Packets | 0 | 0 | - | - |
| Zone Name Conflicts | 0 | 0 | - | - |
| Zone Count Conflicts | 0 | 0 | - | - |
| AEP Statistics : | | | | |
| Echo Requests In | 0 | 0 | - | - |
| Echo Replies Out | 0 | 0 | - | - |
| Echo Requests Out | 0 | 0 | - | - |
| Echo Replies In | 0 | 0 | - | - |
| NBP Statistics : | | | | |
| Entity Filter Matches | - | - | - | - |

The elements of this display are described as follows:

DDP Statistics General Datagram Counts

| | |
|--------------------|--|
| Locally Originated | Number of packets transmitted out a port that originated within the router (for example, Routing Table Maintenance Protocol (RTMP) route information packets). |
| Short DDP Out | Number of short Datagram Delivery Protocol (DDP) packets transmitted out a port (always 0, because AppleTalk Phase 2 does not use short DDP headers; present because management information base (MIB) uses the same data structures). |

| | |
|-----------------------|---|
| Long DDP Out | Number of packets transmitted out a port with Long DDP Headers (all non-AppleTalk Address Resolution Protocol (ARP) packets). |
| Total In | Number of packets received on a port by DDP from external devices. |
| In - Not Local Dest | Number of packets received from external devices out port not addressed specifically to this router and that are not broadcast/multicast packets of interest to RTMP, Zone Information Protocol (ZIP) or Name Binding Protocol (NBP) protocols. |
| In - Locally Destined | Number of packets received from external devices out port addressed specifically to this box or that are broadcast/multicast packets of interest to the AppleTalk protocols implemented RTMP, ZIP, NBP, AppleTalk Echo Protocol (AEP) |

Dropped Datagram Counts

| | |
|--------------------|---|
| No Recipient | Number of packets received on port destined for AppleTalk node on router for protocols not present or ready to accept packets. |
| No Route | Number of packets received on port not destined for this router for which no route was found in the AppleTalk routing table. |
| Data Too Short | The total number of input DDP datagrams dropped because the received data length was less than the data length specified in the DDP header or the received data length was less than the length of the expected DDP header. |
| Data Too Long | The total number of input DDP datagrams dropped because the received data length was greater than the data length specified in the DDP header or because they exceeded the maximum DDP datagram size. |
| Broadcast Error | The total number of input DDP datagrams dropped because this entity was not their final destination and they were addressed to the link level broadcast. |
| Short Header Error | The total number of input DDP datagrams dropped because this error was not their final destination and their type was short DDP (always 0 since AppleTalk Phase 2) |
| Hop Count Error | The total number of input DDP datagrams dropped because this entity was not their final destination and their hop count would exceed 15. |
| Checksum Error | The total number of input DDP datagrams dropped because of a checksum error. |

RTMP Statistics

| | |
|-------------------|---|
| Network Filter | Number of packets not routed through port because of a Network Number |
| Matches | Filter in effect. |
| Data/Responses In | Number of RTMP Data packets or Route Data Request responses received over port. |

| | |
|-------------------------|---|
| Data/Responses Out | Number of RTMP Data packets broadcast or Route Data Request responses sent out port. |
| Requests In | Number of RTMP Request packets received over port. |
| Requests Out | Number of RTMP Request packets sent out port. |
| Route changes (dist) | Number of times RTMP changes the Next Internet Router in a routing entry because the hop count advertised in a routing tuple was equal to the current hop count for a particular network. |
| Route changes (shorter) | Number of times RTMP changes the Next Internet Router in a routing entry because the hop count advertised in a routing tuple was less than the current hop count for a particular network. |
| Network Dist. Exceeded | Number of times RTMP deletes a route from the table because of a distance change that makes the network inaccessible because of the 15-hop limitation. |
| Network Route Deletes | Number of times RTMP deletes a route because it was aged out of the table. This can help to detect routing problems. |
| Invalid Packets | Number of packets ignored by RTMP because of invalid data found within the RTMP data, such as invalid network numbers. SHow DIAGnostics should be checked if nonzero values are seen. |
| Bad Tuple Packets | Number of routing information packets received over the port containing Invalid Data Tuples. SHow DIAGnostics should be checked if nonzero values are seen. |
| Net Number Overlaps | Number of network overlaps detected between routing information obtained from another router over the given port and current network entries in the routing table. SHow DIAGnostics should be checked if nonzero values are seen. |

ZIP Statistics

| | |
|-----------------------|---|
| Queries In | Number of ZIP query packets received over the port. |
| Queries Out | Number of ZIP query packets sent to other routers over the port. |
| Replies In | Number of ZIP query response packets received over the port from other routers. |
| Replies Out | Number of extended ZIP query response packets transmitted out the port. |
| Extended Replies In | Number of ZIP Extended Replies received by this entity. |
| Extended Replies Out | Number of ZIP Extended Replies sent by this entity. |
| GetZoneList Req. In | Number of ZIP GetZoneList transactions received by this entity. |
| GetZoneList Rep. Out | Number of ZIP GetZoneList transactions sent by this entity. |
| GetLocalZones Req. In | Number of ZIP GetLocalZones transactions received by this entity. |

| | |
|------------------------|---|
| GetLocalZones Rep. Out | Number of ZIP GetLocalZonesReply transactions sent by this entity. |
| GetMyZone Req. In | Number of ZIP GetMyZone transactions received by this entity. |
| GetMyZone Rep. Out | Number of ZIP GetMyZoneReply transactions sent by this entity. |
| GetNetInfo Req. In | Number of ZIP GetNetInfo packets received by this entity. |
| GetNetInfo Rep. Out | Number of ZIP GetNetInfoReply packets sent by this entity. |
| GetNetInfo Req. Out | Number of ZIP GetNetInfo packets sent by this entity. |
| GetNetInfo Rep. In | Number of ZIP GetNetInfoReply packets received by this entity. |
| Invalid Packets | Number of ZIP packets of all types received with invalid information detected. SHow DIAGNOSTICS should be checked if nonzero values are seen. |
| Zone Name Conflicts | SHow DIAGNOSTICS should be checked if nonzero values are seen. |
| Zone Count Conflicts | SHow DIAGNOSTICS should be checked if nonzero values are seen. |

AEP Statistics

| | |
|-------------------|---|
| Echo Requests In | Number of AppleTalk Echo requests received. |
| Echo Replies Out | Number of AppleTalk Echo replies sent. |
| Echo Requests Out | Number of AppleTalk Echo requests sent. |
| Echo Replies In | Number of AppleTalk Echo replies received. |

NBP Statistics

| | |
|-----------------------|--|
| Entity Filter Matches | Number of times a Name Binding protocol lookup or reply packet is ignored because it matches an entity filter. |
|-----------------------|--|

ARP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -ARP command:

```

ACCUMULATED VALUES
===== ARP statistics =====
Data Pkts Discarded
  Aged                0
  Queue Full         0
  Addr List Full     0
Data Pkts In Queue   0

== ARP statistics == 1== 2== 4==
Requests Received:
  All Requests Rcvd   627273  -  -
  All Requests Rspd   54      -  -
  Proxy Requests Rcvd 625489  -  -
  Proxy Requests Rspd 0        -  -
  Agent Requests Rcvd 0        -  -
  Agent Requests Rspd 0        -  -
    
```

| | | | |
|--------------------------|----|---|---|
| Requests Sent: | | | |
| All Req Sent | 17 | - | - |
| Repeat Req Sent | 0 | - | - |
| Refresh Req Sent | 12 | - | - |
| InARP Statistics: | | | |
| InARP Request(out) | 0 | - | - |
| InARP Response(in) | 0 | - | - |
| InARP Request(in) | 0 | - | - |
| InARP Response(out) | 0 | - | - |
| Pkts Discarded | 0 | - | - |
| RARP Statistics: | | | |
| Outgoing Requests | 0 | - | - |
| Incoming Responses | 0 | - | - |
| Incoming Requests | 15 | - | - |
| Outgoing Responses | 0 | - | - |

The elements of this display are described as follows:

Data Pkts Discarded

| | |
|----------------|---|
| Aged | Number of packets discarded on a port as a result of time-outs waiting for a response. |
| Queue Full | Number of packets discarded on a port as a result of a full routing queue. The maximum allowable number of packets waiting for Internet address resolution is 10. |
| Addr List Full | Number of packets discarded on a port as a result of a full address list. The maximum allowable number of addresses waiting for resolution is 10. |

Data Pkts In Queue

Number of packets on a port still in the queue waiting for address resolution.

Requests Received

| | |
|---------------------|---|
| All Requests Rcvd | Number of Internet address resolution requests received on a port. |
| All Requests Rspd | Number of responses to Internet address resolution requests on a port. |
| Proxy Requests Rcvd | Number of proxy requests received on a port for an Internet address not on the network where the request originated. |
| Proxy Requests Rspd | Number of responses to proxy requests responded to on a port. |
| Agent Requests Rcvd | Number of agent requests received on a port. An agent is a designated router that can respond to Address Resolution Protocol (ARP) requests for a PC; for example, when the ARP Service is not implemented. |
| Agent Requests Rspd | Number of responses to agent requests on a port. |

Requests Sent

| | |
|-------------------|--|
| All Requests Sent | Number of Internet address resolution requests sent on a port since the last boot or since dynamic information in routing tables was last flushed. |
|-------------------|--|

| | |
|-------------------|--|
| Repeat Req. Sent | Number of Internet address resolution repeat requests sent on a port since the last boot or since dynamic information in routing tables was last flushed. |
| Refresh Req. Sent | Number of Internet address resolution refresh requests sent on a port since the last boot or since dynamic information in routing tables was last flushed. |

InARP Statistics

| | |
|----------------------|---|
| InARP Request (out) | Number of InARP requests sent on a port. |
| InARP Response (in) | Number of InARP responses received on a port. |
| InARP Request (in) | Number of InARP requests received on a port. |
| InARP Response (out) | Number of InARP responses sent on a port. |
| Pkts Discarded | Number of packets discarded on a port. |

RARP Statistics

| | |
|--------------------|--|
| Outgoing Requests | Number of outgoing Reverse Address Resolution Protocol (RARP) requests transmitted on a port by a RARP client. |
| Incoming Responses | Number of incoming RARP responses received on a port by a RARP client. |
| Incoming Requests | Number of incoming RARP requests received on a port by the RARP server. |
| Outgoing Responses | Number of outgoing RARP responses transmitted on a port by the RARP server. |

ATUN Service

The following is an example of the display generated by the `SHoW -SYS STATISTICS -ATUN` command:

```

ACCUMULATED VALUES
== ATUN statistics ===== 1===== 2===== 3===== 4=====
RCVD: Address Pkts          -          -          0          9
    Bytes                   -          -          0         18
    Broadcast Pkts          -          -          0          0
    Bytes                   -          -          0          0
    No CU Pkts              -          -          0          4
    Bytes                   -          -          0          8
Err:Too Short              -          -          0          0
    Parity                  -          -          0          0
    Break                   -          -          0          0
    Framing                 -          -          0          0
    CD Lost                 -          -          0          0
    Internal                -          -          0          0
Xmit: Pkts sent            -          -          0          9
    Bytes Sent              -          -          0         54
    Err: FlowControl        -          -          0          0
  
```

The elements of this display are described as follows:

- Addressed** The packet and byte count for data received from the port and addressed to a specific CU.
- Broadcast** The packet and byte count for data received from the port and addressed to all CUs, either via explicit broadcast or because addressing on the port is disabled.

| | |
|-------------|---|
| No CU | The packet and byte count for frames discarded when addressing is used, and no CU configuration can be found matching the address in the frame. |
| Err | Provides a breakdown count of various error frames for packet counts only. Error frames are discarded. |
| Too Short | Addressing configured on the port, but the framer received was too short to contain an address byte at the configured AddrLOCation value. |
| Parity | This frame was terminated due to receipt of a parity error on the asynch line. |
| Break | This frame was terminated due to receipt of a break error on the asynch line. |
| Framing | This frame was terminated due to receipt of a framing error on the asynch line. |
| CD Lost | This frame was terminated because the DCD control signal dropped during receipt of a character. |
| Internal | This frame was terminated because some internal error (such as a buffer overflow) occurred. |
| Xmit | Shows the packet and byte count of data transmitted to the port by the CU tunnels. |
| FlowControl | Shows the count of packets discarded on transmission due to transmit overflow on the port. |

BGP Service

The following is an example of the display generated by the SHow -SYS STATistics -BGP command:

```

=====BGP Statistics=====
BGP Received:  Messages      Updates      Keepalives    Notification    Bytes      Routes
                0                0                0                0                0          0
                Unreachables  Duplicates
                0                0
BGP Transmitted: Messages      Updates      Keepalives    Notification    Bytes      Routes
                 0                0                0                0                0          0
                 Unreachables  Duplicates
                 0                0
    
```

These are some of the statistics that will be maintained by the BGP Service.

BGP Statistics for All Peers

The following statistics are for the entire router:

| | |
|-----------------------|-------------------------|
| Bytes in | Keepalives Out |
| Bytes out | Networks |
| Updates In | AS paths |
| Updates Out | Unreachables in |
| Notification Messages | Unreachables out |
| Notifications Out | Network Policy discards |
| Keepalives in | AS policy discards |

Per-peer Statistics

InUpdates
 OutUpdates
 InMessages
 OutMessages

BRidge Service

The following is an example of the display generated by the `SHoW -SYS STATISTICS -BRidge` command:

```

ACCUMULATED VALUES
== BRIDGE statistics ==      ==== 1====   ==== 3====   === 5====   === 7====
Bridge Statistics
InFrames                    787          0          -          -
InDiscards                  681          0          -          -
OutFrames                    0           106         -          -
OutDiscards                  0            0          -          -
MtuDiscards                  0            0          -          -
BCLDiscards                  0            0          -          -
BCLInvoked                   0            0          -          -
IPFragmented                 0            0          -          -
IPFragments                  0            0          -          -

```

The elements of this display are described as follows:

| | |
|--------------|--|
| InFrames | Number of good incoming frames. |
| InDiscards | Number of incoming discarded frames. |
| OutFrames | Number of outgoing good frames. |
| OutDiscards | Number of outgoing discarded frames. |
| MtuDiscards | Number of packets discarded as a result of exceeding maximum packet size. Mixed media configurations only. |
| BCLDiscards | Number of packets discarded by the broadcast limit mechanism. |
| BCLInvoked | Number of timer intervals in which the broadcast limit threshold was exceeded. |
| IPFragmented | Total number of IP packets fragmented. |
| IPFragments | Total number of IP fragments generated. |

While InFrames and InDiscards counters are accurate, OutFrames and OutDiscards counters report the number of packets the bridge tried to forward. If the packets are dropped for other reasons after the bridge tried to forward it, these packets will not show up in the OutDiscards counters, but in the port statistics. InDiscards are a normal condition (that is, the destination media access control (MAC) address is on the same network segment as the bridge from which it was received).

If the display indicates that there are very few InDiscards compared to OutFrames on a LAN port, the LAN segments connected by your bridge may not be evenly distributed. The bridge routinely forwards all or most of the frames received on that port. If the display indicates that there is an excessively large number of OutDiscards on a LAN port, the port may be highly saturated, the output port may not be forwarding, or the `-BRidge BroadCastLimit` parameter may not be appropriately set.

BSC Service

The following is an example of the display generated by the SHow -SYS STATistics -BSC command for receive statistics (the display for transmit statistics is similar except for the line heading " Rcvd General Poll"):

```

ACCUMULATED VALUES
== BSC statistics =====      === 1===   === 2===   === 3===   === 4===
Rcvd General Poll                0          0          0          0
  Specific Poll                   0          0          0          0
  Selection                       0          0          0          0
  SOH Data Block                  0          0          0          0
  Data Block - ETB                0          0          0          0
  Data Block - ETX                0          0          0          0
  Data Block - ITB                0          0          0          0
  Trans Data - ETB                0          0          0          0
  Trans Data - ETX                0          0          0          0
  Trans Data - ITB                0          0          0          0
  Bytes                           0          0          0          0
  ACK 0                           0          0          0          0
  ACK 1                           0          0          0          0
  ENQ                             0          0          0          0
  EOT                             0          0          0          0
  NAK                             0          0          0          0
  RVI
  TTD
  WACK
  Unknown

```

The elements of this display are described as follows:

| | |
|--------------------------|---|
| Rcvd (Xmit) General Poll | General polls received (or transmitted). |
| Specific Poll | Specific polls received (or transmitted). |
| Selection | Selections received (or transmitted). |
| SOH Data Block | Data blocks received (or transmitted) with " Start of Header" in block. |
| Data Block - ETB | Data blocks received (or transmitted) with " End of Text Block" in block. |
| Data Block - ETX | Data blocks received (or transmitted) with " End of Text" in block. |
| Data Block - ITB | Data blocks received (or transmitted) with " Intermediate Transmission Block" in block. |
| Trans Data - ETB | Transparent data blocks received (or transmitted) with " End of Text Block" in block. |
| Trans Data - ETX | Transparent data blocks received (or transmitted) with " End of Text" in block. |
| Trans Data - ITB | Transparent data blocks received (or transmitted) with " Intermediate Transmission Block" in block. |
| Bytes | Total number of bytes received (or transmitted) that include only data frames such as " Trans Data - ETB" and excludes all control frames such as " ACK O." |
| ACK 0 | Positive acknowledgment for multipoint selection, point-to-point line bids and even number data blocks. |
| ACK 1 | Positive acknowledgment for odd number data blocks. |
| ENQ | Enquiry. |

| | |
|---------|---|
| EOT | End of Transmission. |
| NAK | Negative acknowledgment |
| RVI | Reverse interrupt. |
| TTD | Temporary Text Delay |
| WACK | Wait before Transmit Positive Acknowledgment
" temporarily not ready to receive." |
| Unknown | Unknown blocks, block is discarded and not transmitted to DLSw. These blocks are not included in the transmit statistics as they have been discarded by the receiver. |

CLNP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -CLNP command:

```

ACCUMULATED VALUES
== CLNP statistics =====      === 1===   === 2===   === 3===   === 4===
Rcvd: good PDU                   0          0          0          0
   pass to client                 0          0          0          0
   bad PDU syntax                 0          0          0          0
   dest unreachable              0          0          0          0
   cksum error                   0          0          0          0
   lifetime expired              0          0          0          0
Xmit: good PDU                   0          0          0          0
   xmit error                     0          0          0          0

```

The elements of this display are described as follows:

CLNP statistics Rcvd: good PDU

Number of protocol data units (PDUs) received.

pass to client Number of PDUs passed to the local client of Connectionless Network Protocol (CLNP).

bad PDU syntax Number of errors caused by PDUs with incorrect syntax.

dest unreachable Number of errors generated because the router cannot forward the PDU.

cksum error Number of checksum errors.

lifetime expired Number of errors caused by expiration of the time-to-live (TTL) field in the PDU.

Xmit: good PDU

Number of PDUs transmitted successfully.

Xmit error Number of errors caused by queue overflow.

DECnet Service

The following is an example of the display generated by the SHow -SYS STATistics -DECnet command, showing per-port statistics:

```

ACCUMULATED VALUES
=== DECnet statistics ===== 1===== 2===== 3=====
Data Messages
  Received                0          -          -
  MaxVisits Exceeded      0          -          -
  This Node               0          -          -
  No Route                0          -          -
  Bad version             0          -          -
  Transmitted             0          -          -
Routing Messages
  Level 1 Received        0          -          -
  Level 2 Received        0          -          -
  Bad Checksum            0          -          -
  Level 1 Transmitted     0          -          -
  Level 2 Transmitted     0          -          -
Hello Messages
  Received                0          -          -
  End Node                0          -          -
  Discarded               0          -          -
  Transmitted             0          -          -
Phase V Data Messages
  Received                0          -          -
  No Phase IV route       0          -          -
  Transmitted             0          -          -
  No Phase V route       0          -          -
Internetwork Data Messages
  INR Transmitted        0          -          -
  ATG Transmitted        0          -          -

```

The elements of this display are described as follows:

Data Messages

| | |
|--------------------|--|
| Received | Number of data packets received. |
| MaxVisits Exceeded | Number of packets that exceed the maximum number of visits allowed by the MaxVisits parameter. These packets are assumed to be looping and are therefore discarded. |
| This Node | Number of packets whose destination node is the router itself and are therefore discarded. |
| No Route | Number of packets discarded because the router has no information (routes) available for routing them. |
| Bad version | Number of DECnet protocol packets encoded with a version number that is not supported by the 3Com implementation, for example, DECnet Phase III packets. Packets encoded in short or invalid format are also included. |
| Transmitted | Number of packets sent. |

Routing Messages

| | |
|---------------------|--|
| Level 1 Received | Number of Level 1 routing messages received. |
| Level 2 Received | Number of Level 2 routing messages received. |
| Bad Checksum | Number of packets received with checksum errors. |
| Level 1 Transmitted | Number of Level 1 routing messages sent. |
| Level 2 Transmitted | Number of Level 2 routing messages sent. |

Hello Messages

| | |
|-------------|---|
| Received | Number of hello messages received. |
| End Node | Number of end node hello messages received. |
| Discarded | Number of hello messages discarded. |
| Transmitted | Number of hello messages sent. |

Phase V Data Messages

| | |
|-------------------|--|
| Received | Number of Phase V data messages received and successfully transmitted as Phase I data messages. |
| No Phase IV Route | Number of Phase V data messages discarded because a Phase IV route to destination was not available. |
| Transmitted | Number of Phase IV data messages successfully transmitted as Phase V data messages. |
| No Phase V Route | Number of Phase IV data messages discarded because a Phase V route to destination was not available. |

Internetwork Data Messages

| | |
|-----------------|---|
| INR Transmitted | Number of data packets sent to another directly attached network. |
| ATG Transmitted | Number of data packets sent using the user-defined address maps. |

DLSw Service

The following is an example of the display generated by the SHow -SYS STATISTICS -DLSw command:

```

ACCUMULATED VALUES
===== DLSw statistics =====
CANUREACH:          Xmit          9
                   Rcvd           2
ICANREACH           Xmit           0
                   Rcvd           1
REACH_ACK           Xmit           1
                   Rcvd           0
DGRAMFRAME          Xmit           0
                   Rcvd           0
XIDFRAME            Xmit           8
                   Rcvd           7
CONTACT             Xmit           0
                   Rcvd           1
CONTACTED           Xmit           1
                   Rcvd           0
RESTART_DL          Xmit           0
                   Rcvd           0
DL_RESTARTED        Xmit           0
                   Rcvd           0
INFOFRAME           Xmit          12
                   Rcvd          12
ENTERBUSY           Xmit           0
                   Rcvd           0
EXITBUSY            Xmit           0
                   Rcvd           0
HALT_DL             Xmit           0
                   Rcvd           0
  
```

| | | |
|---------------|------|---|
| DL_HALTED | Xmit | 0 |
| | Rcvd | 0 |
| NETBIOS_NQ | Xmit | 0 |
| | Rcvd | 0 |
| NETBIOS_NR | Xmit | 0 |
| | Rcvd | 0 |
| DATAFRAME | Xmit | 0 |
| | Rcvd | 0 |
| NETBIOS_ANQ | Xmit | 0 |
| | Rcvd | 0 |
| NETBIOS_ANR | Xmit | 0 |
| | Rcvd | 0 |
| HALTDL_NO_ACK | Xmit | 0 |
| | Rcvd | 0 |
| TEST_CIR_REQ | Xmit | 0 |
| | Rcvd | 0 |
| TEST_CIR_RSP | Xmit | 0 |
| | Rcvd | 0 |
| OTHERS | Xmit | 0 |
| | Rcvd | 0 |
| DISCARDED | Xmit | 0 |
| | Rcvd | 0 |

The elements of this display are described as follows:

| | |
|---------------|--|
| CANUREACH | Number of CanUReach Station messages transmitted or received. |
| ICANUREACH | Number of ICanReach Station messages transmitted or received. |
| REACH_ACK | Number of Reach Acknowledgment messages transmitted or received. |
| DGRMFRAME | Number of Datagram Frame messages transmitted or received. |
| XIDFRAME | Number of XID frames transmitted or received. |
| CONTACT | Number of Contact Remote Station messages transmitted or received. |
| CONTACTED | Number of Remote Station Contacted messages transmitted or received. |
| RESTART_DL | Number of Restart Data Link messages transmitted or received. |
| DL_RESTARTED | Number of Data Link Restarted messages transmitted or received. |
| INFOFRAME | Number of Information (I) Frame messages transmitted or received. |
| ENTERBUSY | Number of Enter Link Station Busy messages transmitted and received. |
| EXITBUSY | Number of Exit Link Station Busy messages transmitted or received. |
| HALT_DL | Number of Halt Data Link messages transmitted or received. |
| DL_HALTED | Number of Data Link Halted messages transmitted or received. |
| NETBIOS_NQ | Number of NetBIOS Name Query messages transmitted or received. |
| NETBIOS_NR | Number of NetBIOS Name Recognized messages transmitted or received. |
| DATAFRAME | Number of Dataframe messages transmitted or received. |
| NETBIOS_ANQ | Number of NetBIOS Add Name Query messages transmitted or received. |
| NETBIOS_ANR | Number of NetBIOS Add Name Response messages transmitted or received. |
| HALTDL_NO_ACK | Number of Halt Data Link No Acknowledgment messages transmitted or received. |
| TEST_CIR_REQ | Number of Test Circuit Request messages transmitted or received. |
| TEST_CIR_RSP | Number of Test Circuit Response messages transmitted or received. |
| OTHERS | Number of messages undefined in RFC 1434 transmitted or received. |
| DISCARDED | Number of Discarded messages transmitted or received. |

DVMRP Service

The following is an example of a display generated by the `SHoW -SYS STATistics -DVMRP` command:

```

ACCUMULATED VALUES
== DVMRP statistics =====
Rcvd from MOSPF                0
Sent to MOSPF                  0
Pruned by MOSPF                0
== DVMRP statistics =====   === 1===   === 2===   === 3===   === 4===
Pkts Received (total)          0           0           2011         0
  Reports                      0           0           12966        0
  Prunes                       0           0            667         0
  Grafts                       0           0             0         0
  Graft Acks                    0           0             0         0
Pkts Transmitted                0           0             0         0
  Reports                    12959         12959             0         0
  Prunes                     0             0             0         0
  Grafts                     0             0             0         0
  Graft Acks                  0             0             0         0
Pkts Forwarded                 5960K         671             0         0
Pkts Discarded                 0             0             0         0
  NoRoute                     0             0             0         0
  WrongPort                   0             0             0         0
  Unknown Type                 0             0             0         0
  MiscErr                     0             0             0         0
IP over IP Statistics:         0             0             0         0
  Pkts Received                0             0             0         0
  Pkts Discarded               0             0             0         0

```

The elements of this display are described as follows:

DVMRP Statistics

Rcvd from MOSPF Total number of packets received from Multicast Open Shortest Path First Protocol (MOSPF) domains.

Sent to MOSPF Total number of packets transmitted to MOSPF domains.

Pruned by MOSPF Total number of packets pruned by MOSPF domains for no listener in the MOSPF domains.

Pkts Received

Total number of DVMRP packets received.

Reports Number of route update packets received.

Prunes Number of Prune packets received from a downstream neighbor router.

Grafts Number of Graft packets received from a downstream neighbor router.

Graft Acks Number of Graft Acknowledge packets received from the upstream parent router.

Pkts Transmitted

Total number of DVMRP packets transmitted.

Reports Number of route update packets transmitted.

Prunes Number of Prune packets transmitted to the upstream parent router.

Grafts Number of Graft packets transmitted to the upstream parent router.

Graft Acks Number of Graft Acknowledge packets transmitted to any downstream neighbor router.

Pkts Forwarded Total number of packets forwarded.

Pkts Discarded

NoRoute Number of packets received when there is no route to the source.
 WrongPort Number of packets received on a port that is not used to forward to the source.
 Unknown Type Number of packets received that are of unknown type.
 MiscErr Number of packets received when the system is out of resources.

IP over IP Statistics

Pkts Received Total number of IP-over-IP packets received.
 Pkts Discarded Total number of IP-over-IP packets discarded.

FR Service

The following is an example of the display generated by the SHow -SYS STATISTICS -FR command:

```

ACCUMULATED VALUES
== FR statistics =====      === 1===   === 3===   === 5===   === 7===
Frame Relay Port Statistics:
LMI Frames Xmit                0         0         0         0
LMI Frames Recv                0         0         0         0
Invalid DLCI frames            0         0         0         0
Small Frames Recv              0         0         0         0
Port Status Change             0         0         0         0
    
```

The elements of this display are described as follows:

Frame Relay Port Statistics

LMI Frames Xmit Number of messages successfully transmitted.
 LMI Frames Recv Number of messages successfully received.
 Invalid DLCI frames Number of DLCI frames invalid.
 Small Frames Recv Number of small frames successfully received.
 Port Status Change Number of port status changes.

IDP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -IDP command:

```

ACCUMULATED VALUES
== IDP statistics =====      === 1===   === 3===   === 5===   === 7===
IDP Statistics:
Received                        0         0         0         0
Forwarded                       0         0         0         0
Passed to client                 0         0         0         0
Xmitted                          0         0         0         0
Discarded                        0         0         0         0
    
```

The elements of this display are described as follows:

IDP Statistics

Received Indicates the number of packets received on a port since boot-up time or the last flushing. This number is the total number of packets received from the network including Forwarded packets, Broadcast and Unicast packets addressed to the router and successfully delivered to initial domain identifier (IDP) clients, and some Discarded packets.

| | |
|------------------|--|
| Forwarded | Indicates the number of packets routed successfully to other ports since boot-up time or the last flushing. Those packets generated by the router itself are not included in this category. |
| Passed to client | Indicates the number of broadcast packets or unicast packets addressed to the router and successfully delivered to proper IDP clients. On the router, only Xerox Network Systems (XNS) Routing Information Protocol (RIP) and partitioned emulation programming (PEP) clients reside. So this number is the total number of RIP or PEP packets received on the port since boot-up time or the last flushing. |
| Xmitted | Indicates the number of packets generated and transmitted by the router since boot-up time or the last flushing. There can be only two types of packets (RIP and PEP) generated by the router. |
| Discarded | Indicates the number of packets discarded by IDP because of various errors such as bad framed packets, packets without any data, packets destined to other networks when IDP routing is turned off, packets addressed to the router but no clients available, etc. This number includes bad packets received from either networks or IDP clients. |

IP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -IP command:

```
== IP statistics ===== 1===== 2===== 4=====
```

Datagram Rates(pkts/s):

| | | | |
|---------------|---|---|---|
| Rcvd from Net | 0 | 0 | 0 |
| Txmit to Net | 0 | 0 | 0 |

IP Datagrams :

NORMAL PATH

Totals -

| | | | |
|-----------------------|-----|-----|---|
| Rcvd from Net | 850 | 404 | 0 |
| Rcvd and Fwd | 1 | 0 | 0 |
| Rcvd by Local Client | 261 | 6 | 0 |
| Txmit to Net | 12 | 6 | 0 |
| Txmit by Forwarding | 1 | 0 | 0 |
| Txmit by Local Client | 11 | 6 | 0 |

Unicasts -

| | | | |
|-----------------------|-----|---|---|
| Rcvd from Net | 449 | 2 | 0 |
| Rcvd and Fwd | 1 | 0 | 0 |
| Txmit to Net | 5 | 0 | 0 |
| Txmit by Forwarding | 1 | 0 | 0 |
| Txmit by Local Client | 4 | 0 | 0 |

Multicasts -

| | | | |
|-----------------------|---|---|---|
| Rcvd from Net | 0 | 0 | 0 |
| Rcvd and Fwd | 0 | 0 | 0 |
| Txmit to Net | 6 | 6 | 0 |
| Txmit by Forwarding | 0 | 0 | 0 |
| Txmit by Local Client | 6 | 6 | 0 |

Broadcasts -

| | | | |
|-------------------------|-----|-----|---|
| Rcvd from Net | 401 | 402 | 0 |
| Rcvd and Fwd | 0 | 0 | 0 |
| Txmit to Net | 1 | 0 | 0 |
| Txmit by Forwarding | 0 | 0 | 0 |
| Txmit by Local Client | 1 | 0 | 0 |
| OPTIMIZED PATH - | | | |
| Rcvd and Fwd | 0 | 0 | 0 |
| Txmit by Forwarding | 0 | 0 | 0 |
| Multicasts - | | | |
| Rcvd from Net | 0 | 0 | 0 |
| Rcvd and Fwd | 0 | 0 | 0 |
| Txmit by Forwarding | 0 | 0 | 0 |
| IP Fragmentation: | | | |
| Datagrams Fragmented | 0 | 0 | 0 |
| Fragments Generated | 0 | 0 | 0 |
| Fragmentation Failures | 0 | 0 | 0 |
| Fragments Received | 0 | 0 | 0 |
| Datagrams Assembled | 0 | 0 | 0 |
| Reassembly Failures | 0 | 0 | 0 |
| Errors: | | | |
| Filtering Discards | 0 | 0 | 0 |
| Rcvd Bad Header | 0 | 0 | 0 |
| Rcvd Bad IP Addr | 0 | 0 | 0 |
| Rcvd Unknown Proto | 0 | 0 | 0 |
| Other Receive Errs | 143 | 420 | 0 |
| Route Lookup Failed | 2 | 2 | 0 |
| Invalid IP option | 0 | 0 | 0 |
| TTL expired | 0 | 0 | 0 |
| Buffer Error | 0 | 0 | 0 |
| Other Transmit Errs | 0 | 0 | 0 |
| ICMP Messages (totals): | | | |
| Received Messages | 0 | 0 | 0 |
| Messages Discarded | 0 | 0 | 0 |
| Transmit Messages | 8 | 6 | 0 |
| Transmit Failures | 0 | 0 | 0 |
| ICMP Queries : | | | |
| Echo Txmit | 0 | 0 | 0 |
| Echo Rcvd | 0 | 0 | 0 |
| Addr Mask Txmit | 0 | 0 | 0 |
| Addr Mask Rcvd | 0 | 0 | 0 |
| RDP Solicits Txmit | 0 | 0 | 0 |
| RDP Solicits Rcvd | 0 | 0 | 0 |
| ICMP Responses : | | | |
| Echo Reply Txmit | 0 | 0 | 0 |
| Echo Reply Rcvd | 0 | 0 | 0 |
| Addr Mask Reply Txmit | 0 | 0 | 0 |
| Addr Mask Reply Rcvd | 0 | 0 | 0 |
| RDP Advertise Txmit | 6 | 6 | 0 |
| RDP Advertise Rcvd | 0 | 0 | 0 |
| ICMP Redirects : | | | |
| Redirects Txmit | 2 | 0 | 0 |

| | | | |
|---------------------|---|---|---|
| Redirects Rcvd | 0 | 0 | 0 |
| ICMP Dest Unreach : | | | |
| Dest Unreach Txmit | 0 | 0 | 0 |
| Dest Unreach Rcvd | 0 | 0 | 0 |
| ICMP TTL Msgs : | | | |
| Time Exceed Txmit | 0 | 0 | 0 |
| Time Exceed Rcvd | 0 | 0 | 0 |

To see the IP statistics for a particular port, use:

```
SHow [!port] STAT -IP
```

For example, to see the statistics for port 4, enter:

```
SHow !4 STAT -IP
```

IP Statistics Descriptions

Datagram Rates (pkts/s) Rcvd from Net : The total number of input datagrams received per second on a port including those received in error.

Txmit to Net : The total number of IP-datagrams transmitted per second to the port.

IP Datagrams

NORMAL PATH

Totals

Rcvd from Net : The total number of input datagrams received on a port including those received in error.

Rcvd and Fwd : The number of input datagrams for which this port was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination.

Rcvd by Local Client : The number of input datagrams passed on to the local client.

Txmit to Net : The number of datagrams transmitted on the network thru this port. The statistics collected here are the sum of the statistics collected in "Txmit by Forwarding" and "Txmit by Local Client" as described below.

Txmit by Forwarding : The total number of IP-datagrams transmitted to the port. Note that this statistics is different from the one below in the sense that it only includes those datagrams which were transmitted to the router by various other routers in the network because this router happens to be either the final destination or the router in the path to the final destination. This counter does not include any datagrams maintained in the "Originated by system" statistics.

Txmit by Local Client : The total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission.

Unicasts

Rcvd from Net : The total number of unicast datagrams received on a port including those received in error.

Rcvd and Fwd : The number of unicast datagrams for which this port was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination.

Txmit to Net : The number of unicast datagrams transmitted on the network thru this port. The statistics collected here are the sum of the statistics collected in "Txmit by Forwarding" and "Txmit by Local Client" as described below.

Txmit by Forwarding : The total number of unicast IP-datagrams transmitted to the port. Note that this statistics is different from the one below in the sense that it only includes those datagrams which were transmitted to the router by various other routers in the network because this router happens to be either the final destination or the router in the path to the final destination. This counter does not include any datagrams maintained in the "Originated by system" statistics.

Txmit by Local Client : The total number of unicast IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission.

Multicasts

Rcvd from Net : The total number of multicast datagrams received on a port including those received in error.

Rcvd and Fwd : The number of multicast datagrams for which this port was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination.

Txmit to Net : The number of multicast datagrams transmitted on the network thru this port. The statistics collected here are the sum of the statistics collected in "Txmit by Forwarding" and "Txmit by Local Client" as described below.

Txmit by Forwarding : The total number of multicast IP-datagrams transmitted to the port. Note that this statistics is different from the one below in the sense that it only includes those datagrams which were transmitted to the router by various other routers in the network because this router happens to be either the final destination or the router in the path to the final destination. This counter does not include any datagrams maintained in the "Originated by system" statistics.

Txmit by Local Client : The total number of multicast IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission.

Broadcasts

Rcvd from Net : The total number of broadcast datagrams received on a port including those received in error.

Rcvd and Fwd : The number of broadcast datagrams for which this port was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination.

Txmit to Net : The number of broadcast datagrams transmitted on the network thru this port. The statistics collected here are the sum of the statistics collected in "Txmit by Forwarding" and "Txmit by Local Client" as described below.

Txmit by Forwarding : The total number of broadcast IP-datagrams transmitted to the port. Note that this statistics is different from the one below in the sense that it only includes those datagrams which were transmitted to the router by various other routers in the network because this router happens to be either the final destination or the router in the path to the final destination. This counter does not include any datagrams maintained in the "Originated by system" statistics.

Txmit by Local Client : The total number of broadcast IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission.

OPTIMIZED PATH

Rcvd and Fwd : The total number of datagrams received through fast path on a port including those received in error.

| | |
|------------------|--|
| | <p>Txmit by Forwarding : The total number of multicast IP-datagrams transmitted to the port thru fast path. Note that this statistics is different from the one below in the sense that it only includes those datagrams which were transmitted to the router by various other routers in the network because this router happens to be either the final destination or the router in the path to the final destination. This counter does not include any datagrams maintained in the "Originated by system" statistics.</p> |
| Multicasts | <p>Rcvd from Net : The total number of multicast datagrams received through fast path on a port including those received in error.</p> <p>Rcvd and Fwd : The number of multicast datagrams receive through the fast path for which this port was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination.</p> <p>Txmit by Forwarding : The total number of multicast IP-datagrams transmitted to the port thru fast path. Note that this statistics is different from the one below in the sense that it only includes those datagrams which were transmitted to the router by various other routers in the network because this router happens to be either the final destination or the router in the path to the final destination. This counter does not include any datagrams maintained in the "Originated by system" statistics.</p> |
| IP Fragmentation | <p>Datagrams Fragmented : The number of IP datagrams that have been successfully fragmented at this port.</p> <p>Fragments Generated : The number of IP datagram fragments that have been generated as a result of fragmentation at this port.</p> <p>Fragmentation Failure : The number of IP datagrams that have been discarded because they needed to be fragmented at this port but could not be, e.g., because their Don't Fragment flag was set.</p> <p>Fragments Received : The number of IP fragments received which needed to be reassembled at this entity.</p> <p>Datagrams Assembled : The number of IP datagrams successfully reassembled.</p> <p>Reassembly Failures : The number of failures detected by the IP reassembly algorithm (for whatever reason: timed out, errors, etc).</p> |
| Errors | <p>Filtering Discards: The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded due to the filter rules e.g. Firewall denying certain IP packets depending on the filter rules configured by the user.</p> <p>Rcvd Bad Header : The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options</p> <p>Rcvd Bad IP Addr : The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this port. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E).</p> <p>Rcvd Unknown Proto : The number of locally-addressed datagrams discarded because of an unknown or unsupported protocol.</p> |

Other Receive Errs : The number of input datagrams discarded because of the errors which do not belong to any of the above category. This counter is incremented under any of the following conditions:

- 1 Received fragmented UDP broadcast datagrams addressed to NB2.
- 2 Received a datagram with the SORuce route option turned on AND RelaySrcRoute is disabled in the IP control field on NB2.
- 3 If a directed broadcast packet is received by the netbuilder and FwdSubnetBcast is set to "noFwdSubnetBcast" in the IP control field.
- 4 Received an unknown type of ICMP packet.
- 5 If the incoming datagram could not be delivered to the local client because of mailbox error (like mailbox was full).
- 6 NB2 received a MAC broadcast packet and the destination IP address in the IP header is not a directed broadcast address (NB2 does not forward MAC broadcast packet)

Route Lookup Failed : The number of IP datagrams discarded because no route could be found to transmit them to their destination.

Invalid IP option : The number of IP datagrams which could not be transmitted because of invalid options in the packet. i.e. the options were either not supported or error occurred while processing the packet.

TTL expired : The number of IP datagrams which could not be transmitted because of invalid (0) value in the TTL field.

Buffer Error : The number of IP datagrams which could not be transmitted because of the buffer unavailability.

Other Transmit Errs : The number of output IP datagrams which could not be transmitted on the network due to any of the following:

- Bad IP header in the transmit packet.
- TTL field is 0
- Invalid option specified
- if it is a broadcast/multicast packet on X.25 link
- the lower layer did not transmit for whatever reasons.

ICMP Messages (totals:

Received Messages : The total number of ICMP packets received on the port.

Messages Discarded : The number of ICMP messages which the port received but determined as having ICMP specific errors (bad ICMP checksums, bad length etc)

Transmit Messages : The total number of ICMP messages transmitted on the port.

Transmit Failures : The number of ICMP messages which this port did not send due to problems discovered within ICMP such as the lack of buffers.

ICMP Queries

Echo Txmit : The number of ICMP echo (request) messages transmitted.

Echo Rcvd : The number of ICMP echo (request) messages received.

Addr Mask Txmit : The number of ICMP Address Mask Request messages transmitted.

| | |
|-------------------|---|
| | Addr Mask Rcvd : The number of ICMP Address Mask Request messages received. |
| | RDP Solicits Txmit : The number of RDP solicitation messages transmitted on the port. |
| | RDP Solicits Rcvd : The number of RDP solicitation messages received on the port. |
| ICMP Response: | Echo Reply Txmit : The number of ICMP echo reply messages transmitted. |
| | Echo Reply Rcvd : The number of ICMP echo reply messages received. |
| | Addr Mask Reply Txmit : The number of ICMP Address Mask Reply messages transmitted on the port. |
| | Addr Mask Reply Rcvd : The number of ICMP Address Mask Reply messages received on the port. |
| | RDP Advertise Txmit : The number of RDP Advertisement messages transmitted on the port. |
| | RDP Advertise Rcvd : The number of RDP Advertisement messages received on the port. |
| ICMP Redirects | Redirects Txmit : The number of ICMP Redirect messages transmitted on the port. |
| | Redirects Rcvd : The number of ICMP Redirect messages received on the port. |
| ICMP Dest Unreach | Des Unreach Txmit : The number of ICMP Destination Unreachable transmitted on the network. |
| | Dest Unreach Rcvd : The number of ICMP Destination Unreachable received |
| ICMP TTL Msgs | Time Exceed Txmit : The number of ICMP time exceeded messages transmitted. |
| | Time Exceed Rcvd : The number of ICMP time exceeded messages received. |

IPX Statistics The following is an example of the display generated by the SHow -SYS STATistics -IPX command:

```

ACCUMULATED VALUES
== IPX statistics =====    === 1===   ===2===   === 3===   === 4===
IPX Statistics:
Received                      0         0         0         0
Forwarded                     0         0         0         0
Passed to client               0         0         0         0
Xmitted                       0         0         0         0
Discarded                      0         0         0         0
IPX SPOOF Statistics:
Watchdog Resp (out)           255        -         25        -

```

The elements of this display are described as follows:

Received Number of packets received on a port since boot-up time or the last flush time. This number is the total number of packets received from the network including Routed packets, Broadcast and Unicast packets addressed to the router and successfully delivered to IPX clients, and some Discarded packets.

Forwarded Number of packets routed successfully to other ports since boot-up time or the last flush time. Those packets generated by the router itself are not included in this category.

Passed to client Number of Broadcast packets or Unicast packets addressed to the router and successfully delivered to proper IPX clients. On the router resides only IPX RIP and IPX SAP clients. So this number is the total number of RIP or SAP packets received on the port since boot-up time or the last flush time.

Xmitted Number of packets generated and transmitted by the router since boot-up time or the last flush time. There can be only two types of packets (RIP and SAP) generated by the router.

Discarded Number of packets discarded by IPX because of various errors such as bad framed packets, packets without any data, packets destined to other networks when IPX routing is turned off, packets addressed to the router but no clients available, etc. This number includes bad packets received from either networks or IPX clients.

IPX SPOOF Statistics

Watchdog Resp (out) Number of NCP KeepAliveResponse packets generated by the bridge/router and sent back out on the port as a result of NCP watchdog spoofing being active.

ISIS Service

The following is an example of the display generated by the SHow -SYS STATISTICS -ISIS command:

```

ACCUMULATED VALUES
===== ISIS statistics =====
ISIS statistics
  PDU format error          0
  Corrupted LSP            0
  L1 LinkStateData overload 0
  L2 LinkStateData overload 0
  AreaAddress dropped      0
  SeqNumber overflow       0
  SeqNumber skipped        0
  Own LSP purged           0

== ISIS statistics =====   === 1===   === 3===   === 5===   === 7===
Adjacency change           0           0           0           0
Adjacency reject           0           0           0           0
Corrupted LSP rcvd        0           0           0           0
L2 DIS changes             0           0           0           0
L1 DIS changes             0           0           0           0
PDU sent                   0           0           0           0
PDU rcvd                   0           0           0           0
ID Length mismatch         0           0           0           0

Authentication
  L1 error                  0           0           0           0
  L2 error                  0           0           0           0
  Hello error               0           0           0           0
  
```

The elements of this display are described as follows:

| | |
|---------------------------|--|
| PDU format error | Number of times an Intermediate System-to-Intermediate System (ISIS) protocol data unit (PDU) with an incorrect format was received. |
| Corrupted LSP | Number of times an link state packet (LSP) with unacceptable format or bad information was received. |
| L1 LinkStateData overload | Number of times this router encountered memory resource problems when trying to store a Level 1 LSP PDU. |
| L2 LinkStateData overload | Number of times this router encountered memory resource overload problems when trying to store a Level 2 LSP PDU. |
| AreaAddress dropped | Too many area addresses in the area, causing a manual area address on the route to be dropped. |
| SeqNumber overflow | The sequence number field in the LSP generated by the router has reached the maximum allowed value (approximately 4 billion), which forces the router to go out of service temporarily. |
| SeqNumber skipped | Number of times another router claims to own an LSP generated by this router, but with a high sequence number. |
| Own LSP purged | Number of times another router has purged an LSP generated by this router. |
| Adjacency change | Number of times the adjacency state with nearby routers has gone into UP or DOWN state. |
| Adjacency reject | Number of times an adjacency is rejected to this router because of mismatch in the area address of the two routers. |
| Corrupted LSP rcvd | Number of times an LSP is received on each interface with an unacceptable format or bad information. |
| L2 DIS changes | Number of times the Level 2 designated intermediate system has changed. |
| L1 DIS changes | Number of times the Level 1 designated intermediate system has changed. |
| PDU sent | Number of ISIS PDUs sent, including hello, CSNP, PSNP, and LSP. |
| PDU rcvd | Number of ISIS PDUs received, including hello, CSNP, PSNP, and LSP. The counter includes packets received with format errors. |
| ID Length mismatch | Number of ISIS PDUs received with mismatched ID length fields. All PDUs are counted (hello, CSNP, PSNP, and LSP). Implementations with mismatched ID lengths cannot interoperate. This implementation supports an ID length of six octets. |
| L1 error | Number of L1, LSP, CSNP, or PSNP PDUs received with mismatched Level 1 password. |
| L2 error | Number of L2, LSP, CSNP, or PSNP PDUs received with mismatched Level 2 password. |
| Hello error | Number of Level 1 or Level 2 hello PDUs received with mismatched hello password. |

LLC2 Service

The following is an example of the display generated by the `sHow -SYS STATistics -LLC2` command:

```

ACCUMULATED VALUES
== LLC2 statistics =====    === 1===    ===== 3===    ===== 5===    === 7===
Test Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
Xid Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
UI-Data Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
Sabme Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
I-Data Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
I-Data Bytes
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
RR Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
RNR Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
Reject Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
Disc Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
UA Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
DM Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
FRMR Frames
  Received                      0          0          0          0
  Transmitted                   0          0          0          0
    
```

The elements of this display are described as follows:

Test Frames

- Received Number of test frames received per port.
- Transmitted Number of test frames transmitted per port.

Xid Frames

- Received Number of Xid frames received per port.
- Transmitted Number of Xid frames transmitted per port.

| | | |
|-----------------------|-------------|--|
| UI-Data Frames | Received | Number of Unnumbered Information frames received per port. These frames are typically sent and received by NetBIOS and LAN Network Manager Logical Link Control (LLC) protocols. |
| | Transmitted | Number of Unnumbered Information frames transmitted per port. These frames are typically sent and received by NetBIOS and LAN Network Manager LLC protocols. |
| Sabme Frames | Received | Number of set asynchronous balanced mode extended frames received per port. |
| | Transmitted | Number of set asynchronous balanced mode extended frames transmitted per port. |
| I-Data Frames | Received | Number of valid I-data frames received per port, not including retransmissions. |
| | Transmitted | Number of valid I-data frames transmitted per port, not including retransmissions. |
| I-Data Bytes | Received | Number of valid I-data bytes received per port, not including MAC address bytes. |
| | Transmitted | Number of valid I-data bytes transmitted per port, not including MAC address bytes. |
| RR Frames | Received | Number of receiver ready frames received per port. |
| | Transmitted | Number of receiver ready frames transmitted per port. |
| RNR Frames | Received | Number of receiver not ready frames received per port. |
| | Transmitted | Number of receiver not ready frames transmitted per port. |
| Reject Frames | Received | Number of reject frames received per port. |
| | Transmitted | Number of reject frames transmitted per port. |
| Disc Frames | Received | Number of disconnect frames received per port. |
| | Transmitted | Number of disconnect frames transmitted per port. |
| UA Frames | Received | Number of unnumbered acknowledgment frames received per port. |
| | Transmitted | Number of unnumbered acknowledgment frames transmitted per port. |
| DM Frames | Received | Number of disconnect mode frames received per port. |
| | Transmitted | Number of disconnect mode frames transmitted per port. |
| FRMR Frames | Received | Number of frame reject frames received per port. |
| | Transmitted | Number of frame reject frames transmitted per port. |

MIP Service

The following is an example of a display generated by the SHow -SYS STATISTICS -MIP command:



IGMP is not a service in the user interface. However, you can still obtain IGMP statistics through the MIP Service.

```

ACCUMULATED VALUES
===== MIP statistics =====
Multicast IP datagram          ==1==  ==2==  ==3==  ==4==
Pkts Received (total)         1093   5957K   0       0
  Queries                      0       0       0       0
  Reports                       1078   1620    0       0
  Leaves                        0       0       0       0
Pkts Transmitted              0       0      1943    0
Pkts Discarded (total)        0       0       0       0
  TooShort                     0       0       0       0
  Version Err                   0       0       0       0
  Chksum Err                    0       0       0       0
  Unknown Type                  0       0       0       0

```

The elements of this display are described as follows:

Multicast IP Datagram

Total number of multicast IP datagrams received.

Pkts Received

The total number of Internet Group Management Protocol (IGMP) packets received, including DVMRP packets.

Queries Number of Host Query packets received.

Reports Number of Host Report packets received.

Leaves Number of Host Leave Group packets received.

Pkts Transmitted

Total number of IGMP packets transmitted, including DVMRP packets.

Pkts Discarded

Total number of packets discarded.

TooShort Number of packets received with the data length too short.

Version Err Number of packets received that have a bad version number.

Chksum Err Number of packets received that have bad checksums.

Unknown Type Number of packets received with unknown type.

MOSPF Service

The following is an example of a display generated by the SHow -SYS STATISTICS -MOSPF command:

```

ACCUMULATED VALUES
== MOSPF statistics =====
SPF calculations              18
Cache flushed                 13
Rcvd from DVMRP              152610
Sent to DVMRP                 8
Prune back DVMRP              3
Resource error                0
== MOSPF statistics =====
Receive                       === 1===  === 2===  === 3===  === 4===
  Good                         0       0       0       0
  No Route                     0       0       0       0
  Bad MAC address              0       0       0       0
  Wrong source                  0       0       0       0

```


| | | | |
|---------------------|---|---|---|
| PDU rcvd format err | 0 | 0 | 0 |
| L1 DIS change | 0 | 0 | 0 |
| L2 DIS change | 0 | 0 | 0 |
| Authentication: | | | |
| L1 error | 0 | 0 | 0 |
| L2 error | 0 | 0 | 0 |
| Hello error | 0 | 0 | 0 |

The elements of this display are described as follows:

NLSP statistics

| | |
|---------------------------|---|
| Corrupted LSP | Number of times an LSP with unacceptable format or bad information was received. |
| AreaAddress dropped | Too many area addresses in the area, causing a manual area addresses on the router to be dropped. |
| SeqNumber overflow | The sequence number field in the LSP generated by the router has reached the maximum allowed value (approximately 4 billion), which forces the router to go out of service temporarily. |
| SeqNumber skipped | Number of times another router claims to own an LSP generated by this router, but with a high sequence number. |
| Own LSP purged | Number of times another router has purged an LSP generated by this router. |
| L1 LinkStateData overload | Number of times this router encountered memory resource problems when trying to store a Level 1 LSP PDU. |
| L2 LinkStateData overload | Number of times this router encountered memory resource overload problems when trying to store a Level 2 LSP PDU. |
| Adjacency change | Number of times the adjacency state with nearby routers has gone into UP or DOWN state. |
| Adjacency reject | Number of times an adjacency is rejected to this router because of mismatch in the area address of the two routers. |
| Corrupted LSP rcvd | Number of times an LSP is received on each interface with an unacceptable format or bad information. |
| PDU sent | Number of ISIS PDUs sent, including Hello, CSNP, PSNP, and LSP. |
| PDU rcvd | Number of ISIS PDUs received, including Hello, CSNP, PSNP, and LSP. The counter includes packets received with format errors. |
| PDU rcvd format err | Number of times an ISIS PDU with an incorrect format was received. |
| L1 DIS change | Number of times the Level 1 designated intermediate system has changed. |
| L2 DIS change | Number of times the Level 2 designated intermediate system has changed. |

Authentication

| | |
|-------------|--|
| L1 error | Number of L1, LSP, CSNP, or PSNP PDUs received with mismatched Level 1 password. |
| L2 error | Number of L2, LSP, CSNP, or PSNP PDUs received with mismatched Level 2 password. |
| Hello error | Number of Level 1 or Level 2 Hello PDUs received with mismatched hello password. |

NRIP Service

The following is an example of the display generated by the `SHoW -SYS STATISTICS -NRIP` command:

```

ACCUMULATED VALUES
== NRIP statistics ===== 1===== 2===== 4=====
RIP Updates(out)           0           0           0
RIP Updates(in)            0           0           0
RIP Requests(out)          0           0           0
RIP Requests(in)           0           0           0
RIP Replies(out)           0           0           0
RIP Discarded               0           0           0

```

The elements in the display are described as follows:

NRIP statistics

| | |
|--------------------|--|
| RIP Updates(out) | Number of IPX RIP broadcasts transmitted by the router since the boot time or the last flush time. Both regular updates and triggered updates are included in this category. |
| RIP Updates(in) | Number of IPX RIP broadcasts received on a port by the router. |
| RIP Requests (out) | Number of IPX requests generated by the router. |
| RIP Requests(in) | Number of IPX requests received by the router. |
| RIP Replies(out) | Number of RIP replies generated by the router in response to RIP requests. The number of RIP replies can be bigger than the number of RIP requests depending on the current number of IPX RIP table entries. |
| RIP Discarded | Number of RIP packets dropped by the router because of various errors such as packets received from unknown networks, lost packets, and so forth. |

OSPF Service

The following is an example of the display generated by the SHow -SYS STATISTICS -OSPF command:

```

ACCUMULATED VALUES
===== OSPF statistics =====
SPF calculations          1729
Resource error            0

== OSPF statistics =====  1=====  2=====  3=====  4=====
Hello Rcvd                0          0          0          0
Hello Xmit                 0          7          0          0
DD Rcvd                   0          0          0          0
DD Xmit                    0          0          0          0
LSR Rcvd                   0          0          0          0
LSR Xmit                   0          0          0          0
LSA Rcvd                   0          0          0          0
LSA Xmit                   0          0          0          0
LSU Rcvd                   0          0          0          0
LSU Xmit                   0          0          0          0

Number of DR Election      0          1          0          0
Adjacency UP Events       0          0          0          0
Adjacency DOWN Events     0          0          0          0
Errors:
  Xmit fail                0          0          0          0
  Rcv bad packet header   0          0          0          0
  Mismatch HelloTime      0          0          0          0
  Mismatch RouterDeadTim  0          0          0          0
  Mismatch subnet/mask    0          0          0          0
  Mismatch area ID        0          0          0          0
  Unknown packet type     0          0          0          0
  Authentication Error    0          0          0          0
  Packet Checksum Error   0          0          0          0
  LSA Checksum Error      0          0          0          0
    
```

The elements of this display are described as follows:

OSPF Statistics

- SPF calculations Number of times the router has performed the SPF calculation.
- Resource error Number of times OSPF failed to obtain buffers for packet transmission or LSA storage.
- Hello Rcvd Number of Hello messages received by the router.
- Hello Xmit Number of Hello messages sent by the router.
- DD Rcvd Number of data description packets received by the router.
- DD Xmit Number of data description packets transmitted by the router.
- LSR Rcvd Number of link state information request packets received by the router.
- LSR Xmit Number of link state information request packets sent by the router.
- LSA Rcvd Number of link state acknowledgment packets received by the router.
- LSA Xmit Number of link state acknowledgment packets sent by the router.

| | |
|-----------------------|---|
| LSU Rcvd | Number of link state update packets received by the router. |
| LSU Xmit | Number of link state update packets sent. |
| Number of DR Election | Number of times designated router election has been performed. |
| Adjacency UP Events | Number of times that an adjacency has gone from Down to Up state. |
| Adjacency DOWN Events | Number of times that an adjacency has gone from Up to Down state. |

Errors

| | |
|-------------------------|--|
| Xmit Fail | Number of transmission congestions experienced while transmitting OSPF packets. Congestion happens when the transmit queue overflows, and the OSPF packets are dropped. |
| Rcv bad packet header | Number of errors received by the router because of faulty packet headers. |
| Mismatch HelloTime | Number of Hello packets received with mismatched HelloTimes. In order for two OSPF systems to establish an adjacency, both must have identical HelloTime values. |
| Mismatch RouterDeadTime | Number of Hello packets received with mismatched RouterDeadTimes. In order for two OSPF systems to establish an adjacency, both must have identical RouterDeadTime values. |
| Mismatch subnet/mask | Number of Hello packets received with mismatched subnet or mask. In order for two OSPF systems to establish an adjacency, both must have identical IP subnets and masks. |
| Mismatch area ID | Number of adjacency rejections due to mismatched area ID. In order for two neighbors to become adjacent, they must both be configured with identical OSPF area IDs. |
| Unknown packet type | Number of OSPF packets received that are not one of the known (Hello, DD, LSR, LSA and LSU) packet types. |
| Authentication Error | Number of packets received that fail authentication. |
| Packet Checksum Error | Number of packets received with checksum errors. |
| LSA Checksum Error | Number of link state advertisements with checksum errors. |

PATH Service

The following is an example of the display generated by the SHow -SYS STATISTICS -PATH command. This example also applies for the BRIDGE Service and the PPP Service.

```

ACCUMULATED VALUES
== PATH statistics===== 1===== 2===== 3===== 4=====
Rcvd Packets                9265289  0         0         1297
  Bytes                    2542M    0         0         54488
  Err: CRC                   5         0         0         0
    Framing                  360       0         0         0
    Too Long                  89        0         0         0
    Lost                      0         0         0         0
    Parity                    0         0         0         0
    Break                     0         0         0         0

```

| | | | | |
|----------------------|---------|---|---|---------|
| Xmit Packets | 45445 | 0 | 0 | 55296 |
| Bytes | 3065816 | 0 | 0 | 3135915 |
| Err: Deferred | 779 | 0 | 0 | 0 |
| Collision | 704 | 0 | 0 | 0 |
| Late Collisions | 0 | 0 | 0 | 0 |
| Xcess Collision | 0 | 0 | 0 | 0 |
| Carrier Loss | 0 | 0 | 0 | 0 |
| Underrun | 0 | 0 | 0 | 0 |
| Discard: Buf Overrun | 0 | 0 | 0 | 0 |
| Congestion | 0 | 0 | 0 | 0 |
| Utilization: (%) | 7 | 0 | 0 | 0 |
| Rcv Good: pkt/Sec | 119 | 0 | 0 | 0 |
| Byte/Sec | 32826 | 0 | 0 | 0 |
| Xmit Good: Pkt/Sec | 0 | 0 | 0 | 0 |
| Byte/Sec | 0 | 0 | 0 | 0 |

The elements of this display are described as follows:

Rcvd Packets Number of good packets received on the interface.

| | |
|----------|---|
| Bytes | Number of good bytes received on the specified interface. Includes headers but not cyclic redundancy check (CRC) bytes. |
| Err: CRC | Number of frames that were received but failed the cyclic redundancy check. |
| Framing | Number of frames that were received but were not on a 16-bit Too Long boundary. |
| Too Long | Number of frames discarded because the packet length was longer than the maximum packet length allowed. |
| Lost | Number of receptions aborted because the CPU could not provide the controller chip with memory quickly enough. |
| Parity | Number of asynchronous parity receive errors. |
| Break | Number of asynchronous break receive errors. |

Xmit Packets Number of good packets transmitted.



Failure to terminate your Ethernet network will result in the false detection of transmission on the nonterminated BNC connector. This is due to Ethernet module sensitivity to RF transmissions from nearby boards. If this occurs, the Xmit Packets number displayed in the PATH Service statistics will be incorrect.

| | |
|-----------------|--|
| Bytes | Number of bytes transmitted from a port. Includes headers but not CRC bytes. |
| Err: Deferred | Number of frames that could not be transmitted because of existing traffic on the link. Transmission would have resulted in a collision. A later attempt was made to transmit the frame. |
| Collision | Number of frames that experienced a collision during the first attempt to transmit. |
| Late Collisions | Number of frames that received a collision outside of the preamble. |
| Xcess Collision | Number of frames not discarded after 16 consecutive collisions. |
| Carrier Loss | Number of frames that experienced a loss of the carrier signal during transmission. |

Underrun Number of transmissions aborted because the CPU could not provide the controller chip with data fast enough.

Discard

Buf Overrun Number of good frames lost because of memory overrun. This occurs when the system does not have enough memory to transfer the packet internally for further processing.

Congestion Number of frames that could not be transmitted because of transmit queue overflow.

Utilization: The percentage of time the carrier sense signal was active during the specified interval. The percentage of utilization displayed for HSS ports is based on a full-duplex link. For example, a 64 Kbps circuit can transmit and receive 64 Kbps simultaneously. If this link were transmitting at 64 Kbps and receiving nothing, the percentage of utilization would be 50 percent.

Rcv Good:
 Pkt/Sec Number of good packets received per second.
 Byte/Sec Number of good bytes received per second.

Xmit Good

Pkt/Sec Number of good packets transmitted per second.
 Byte/Sec Number of good bytes transmitted per second.

PORT Service

The following is an example of the display generated by the SHow -SYS STATISTICS -PORT command:

```

ACCUMULATED VALUES
== PORT statistics ===== 1===== 2===== 3===== 4=====
Rcvd : Packets              574285    0         0         282
    Bytes                   299912K   0         0         11858
    Multicast                35752    0         0         282
    Broadcast                5080     0         0         0
Xmit : Packets              9801     0         0         11820
    Bytes                   666413   0         0         670004
    Multicast                722      0         0         8529
    Broadcast                722      0         0         3288
Filter : Custom              0         0         0         0
Discard:
    Buf Overrun              0         0         0         0
    Congestion                0         0         0         0
DialOnDemand Mode:
    DodCallsMade              0         0         0         0
    DodCallsFail              0         0         0         0
    DodUpTime                  0         0         0         0
    DodPktsOut                 0         0         0         0
    
```

The elements of this display are described as follows:

Rcvd

Packets Number of good packets received on a specified port.
 Bytes Number of good bytes received on a specified port.
 Multicast Number of multicast packets received. Multicast packets are sent to more than one station on the network.
 Broadcast Number of broadcast packets received. Broadcast packets are sent to the entire network.

Xmit

Packets Number of good packets sent by ports.
 Bytes Number of good bytes sent by ports.
 Multicast Number of multicast packets sent. Multicast packets are sent to more than one station on the network.
 Broadcast Number of broadcast packets sent. Broadcast packets are sent to the entire network.

Filter

Custom Number of packets that matched the custom filters and were therefore discarded.

Discard

Buffer Overrun Number of packets discarded because of buffer overrun. This occurs when the system does not have enough memory to transfer the packet internally for further processing.
 Congestion Number of packets discarded because of congestion. This occurs when a packet cannot be transmitted within a specified amount of time.

DialOnDemand Mode

DodCallsMade Number of outgoing calls successfully initiated by the port operating in dial-on-demand mode.
 DodCallsFail Number of outgoing calls unsuccessfully initiated by the port operating in dial-on-demand mode.
 DodUpTime Length of time in seconds that the primary path of a port is up while operating in dial-on-demand mode.
 DodPktsOut Number of user-data packets sent out by the port operating in dial-on-demand mode.

PPP Service

The following is an example of the display generated by the SHOW -SYS STATISTICS -PPP command, which displays received and transmitted LCP packets:

```

ACCUMULATED VALUES
== PPP statistics ===== 1===== 3===== 5===== 7=====
LCP path statistics :
Rcvd Conf-Request-      0      -      -      0
  Conf-Ack              -      0      -      0
  Conf-Nak              -      0      -      0
  Conf-Reject          -      0      -      0
  Term-Request         -      0      -      0
  Term-Ack             -      0      -      0
  Code-Reject          -      0      -      0
  Protocol-Reject     -      0      -      0
  Echo-Request        -      20     -      420
  Echo-Reply          -     420    -      420
  Discard-Req         -      0      -      0
  Link Quality Rpt    -      0      -      0
  Unknown Code        -      0      -      0
Xmit Conf-Request-      0      -      -      0
  Conf-Ack              -      0      -      0
  Conf-Nak              -      0      -      0
  Conf-Reject          -      0      -      0
  Term-Request         -      0      -      0
  Term-Ack             -      0      -      0
    
```

| | | | | |
|------------------|---|-----|---|-----|
| Code-Reject | - | 0 | - | 0 |
| Protocol-Reject | - | 1 | - | 0 |
| Echo-Request | - | 428 | - | 428 |
| Echo-Reply | - | 428 | - | 427 |
| Discard-Req | - | 0 | - | 0 |
| Link Quality Rpt | - | 0 | - | 0 |

The elements of this display are described as follows:

LCP path statistics **Rcvd**

| | |
|------------------|---|
| Conf-Request | Number of received LCP packets of code 1 for configure request. |
| Conf-Ack | Number of received LCP packets of code 2 for configure acknowledgment. |
| Conf-Nak | Number of received LCP packets of code 3 for configure negative acknowledgment. |
| Conf-Reject | Number of received LCP packets of code 4 for configure rejection. |
| Term-Request | Number of received LCP packets of code 5 for terminate request. |
| Term-Ack | Number of received LCP packets of code 6 for terminate acknowledgment. |
| Code-Reject | Number of received LCP packets of code 7 for code rejection. |
| Protocol-Reject | Number of received LCP packets of code 8 for protocol rejection. |
| Echo-Request | Number of received LCP packets of code 9 for echo request. |
| Echo-Reply | Number of received LCP packets of code 10 for echo reply. |
| Discard-Req | Number of received LCP packets of code 11 for discard request. |
| Link Quality Rpt | Number of received LCP packets of code 12 for link quality report. |
| Unknown Code | Number of received LCP packets of unknown code. |

Xmit

| | |
|-----------------|--|
| Conf-Request | Number of transmitted LCP packets of code 1 for configure request. |
| Conf-Ack | Number of transmitted LCP packets of code 2 for configure acknowledgment. |
| Conf-Nak | Number of transmitted LCP packets of code 3 for configure negative acknowledgment. |
| Conf-Reject | Number of transmitted LCP packets of code 4 for configure rejection. |
| Term-Request | Number of transmitted LCP packets of code 5 for terminate request. |
| Term-Ack | Number of transmitted LCP packets of code 6 for terminate acknowledgment. |
| Code-Reject | Number of transmitted LCP packets of code 7 for code rejection. |
| Protocol-Reject | Number of transmitted LCP packets of code 8 for protocol rejection. |
| Echo-Request | Number of transmitted LCP packets of code 9 for echo request. |
| Echo-Reply | Number of transmitted LCP packets of code 10 for echo reply. |
| Discard-Req | Number of transmitted LCP packets of code 11 for discard request. |

Link Quality Rpt Number of transmitted LCP packets of code 12 for link quality report.

PPP Over Ethernet Statistics

The following is an example of the display generated by the SHow -SYS STATistics -PPPOE command:

```

===PPPoE Statistics===  ===V1===  ===V2===  ===V3===  ===V4===  ===V5===  ===V6===  ===V7===
PPP Session Packets:
Received                0          0          0          0          0          0          0
Transmitted;            0          0          0          0          0          0          0
Packets Discarded:     0          0          0          0          0          0          0
    
```

Statistics are accumulated and displayed for each virtual port.

RIP IP Service

The following is an example of the display generated by the SHow -SYS STATistics -RIP IP command:

```

ACCUMULATED VALUES
===== RIPIP Statistics =====  1=====  2=====  3=====
RIP/IP Statistics:
Incoming Packets          16347      0          0
  Request Updates         0          0          0
  Response Updates        16347      0          0
  Discarded Updates       0          0          0
Outgoing Packets         16347      0          0
  Request Updates         0          0          0
  Regular Responses       16347      0          0
  Triggered Responses     0          0          0
    
```

The elements of this display are described as follows:

Incoming Packets

- Request Updates Number of incoming RIP request packets on a port. RIP request packets request routing information.
- Response Updates Number of incoming RIP response packets on a port. RIP response packets are sent to convey routing information.
- Discarded Updates Number of update packets discarded on a port because they originated from an unconfigured neighbor.

Outgoing Pkts

- Request Updates Number of outgoing request update packets sent on a port. Request updates are sent either when the router is booted or when RIP has been configured.
- Regular Responses Number of outgoing regular update packets sent on a port. Regular updates are router information packets that are sent out by the router at regular intervals.
- Triggered Responses Number of triggered response packets sent on a port. Triggered responses are router information packets sent out immediately when a network becomes unreachable.

RIPXNS Service

The following is an example of the display generated by the `sHow -SYS STATISTICS -RIPXNS` command:

```

ACCUMULATED VALUES
== RIPXNS statistics ===== 1===== 3===== 5===== 7=====
RIPXNS Statistics
RIP Updates(out)                0          0          0          0
RIP Updates(in)                 0          0          0          0
RIP Requests(out)               0          0          0          0
RIP Requests(in)                0          0          0          0
RIP Replies(out)                0          0          0          0
RIP Discarded                   0          0          0          0

```

The elements of this display are described as follows:

RIP Updates(out) Number of XNS RIP broadcasts transmitted by the router since the boot-time or last flushing. Both regular updates and triggered updates will be included in this category.

RIP Updates(in) Number of XNS RIP broadcasts received on a port by the router.

RIP Requests(out) Number of RIP requests generated by the router.

RIP Requests(in) Number of RIP requests received by the router.

RIP Replies(out) Number of RIP replies generated by the router in response to RIP requests. The number of RIP replies can be bigger than the number of RIP requests in case RIP replies require more than one packet.

RIP Discarded Number of RIP packets dropped by the router because of various errors such as packets received from unknown networks, request packets received when IDP routing is turned off, lost packets, and so forth.

RSVP Service

The following is an example of a display for each port generated by the `SHow STATISTICS -RSVP` command:

```

ACCUMULATED VALUES
== RSVPstatistics =====  =====1  =====2  =====3A  =====3B  =====3C  =====4A  =====4B
RSVP port statistics:
No. of PATH msgs rcvd      0          0          0          81         0          0          0
  RESV msgs rcvd           0          0          0          0          0          0          0
  PATH ERR msgs rcvd       0          0          0          0          0          0          0
  RESV ERR msgs rcvd       0          0          0          0          0          0          0
  PATH TEAR msgs rcvd      0          0          0          0          0          0          0
  RESV TEAR msgs rcvd      0          0          1          0          0          0          0
  CONFIRM msgs rcvd        0          0          0          0          0          0          0
No. of PATH msgs sent      0          81         0          0          0          0          0
  RESV msgs sent           0          0          0          0          0          0          0
  PATH ERR msgs sent       0          0          0          0          0          0          0
  RESV ERR msgs sent       0          0          0          0          0          0          0
  PATH TEAR msgs sent      0          0          0          0          0          0          0
  RESV TEAR msgs sent      0          0          0          0          0          0          0
  CONFIRM msgs sent        0          0          0          0          0          0          0
admission failures         0          0          0          0          0          0          0
other resv failures        0          0          0          0          0          0          0

```

| | | | | | | | |
|---------------------|---|---|--------|--------|---|---|---|
| Resv sent due to sc | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| WF resv w/o scope | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| blockade events | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| resv timeouts | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| path timeouts | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| path/ptear rcv ttl0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| path/ptear snt ttl0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Remaining bw(Bps) | 0 | 0 | 4800 | 245760 | 0 | 0 | 0 |
| Configured bw(Bps) | 0 | 0 | 4800 | 245760 | 0 | 0 | 0 |
| RSVP data pkts sent | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RSVP bytes sent | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Non-RSVP data pkts | 0 | 0 | 148652 | 504479 | 0 | 0 | 0 |

The elements of this display are described as follows:

RSVP Port Statistics

| | |
|------------------------|--|
| No. of PATH msgs rcvd | The number of PATH messages received. |
| RESV msgs rcvd | The number of RESV messages received. |
| PATH ERR msgs rcvd | The number of PATH ERROR messages received. |
| RESV ERR msgs rcvd | The number of RESV ERROR messages received. |
| PATH TEAR msgs rcvd | The number of PATH TEAR messages received. |
| RESV TEAR msgs rcvd | The number of RESV TEAR messages received. |
| CONFIRM msgs rcvd | The number of CONFIRM messages received. |
| No. of PATH msgs sent | The number of PATH messages sent. |
| RESV msgs sent | The number of RESV messages sent. |
| PATH ERR msgs sent | The number of PATH ERROR messages sent. |
| RESV ERR msgs sent | The number of RESV ERROR messages sent. |
| PATH TEAR msgs sent | The number of PATH TEAR messages sent. |
| RESV TEAR msgs sent | The number of RESV TEAR messages sent. |
| CONFIRM msgs sent | The number of CONFIRM messages sent. |
| admission failures | The number of admission failures. |
| other resv failures | The number of other reservation failures. |
| Resv sent due to scope | The number RESV messages sent due to scope. |
| WF resv w/o scope | The number of wild card RESV messages without scope. |
| blockade events | The number of blockade events. |
| resv timeouts | The number RESV timeouts. |
| path timeouts | The number of PATH time outs. |
| path/ptear rcv ttl0 | The number of PATH and PATH TEAR messages received with TimeToLive specification of 0. |
| path/ptear snt ttl0 | The number of PATH and PATH TEAR messages sent with TimeToLive specification of 0. |
| Remaining bw(Bps) | The amount of remaining bandwidth. |
| Configured bw(Bps) | The amount of configured bandwidth. |
| RSVP data pkts sent | The number of RSVP data packets sent. |
| RSVP bytes sent | The number of RSVP bytes sent. |
| Non-RSVP data pkts | The number of non-RSVP data packets. |

SAP Service

The following is an example of the display generated by the `sHow -SYS STATISTICS -SAP` command:

ACCUMULATED VALUES

| == SAP statistics ===== | 1===== | 2===== | 4===== |
|-------------------------|--------|--------|--------|
| SAP Updates(out) | 0 | 0 | 0 |
| SAP Updates(in) | 0 | 0 | 0 |
| SAP Requests(out) | 0 | 0 | 0 |
| SAP Requests(in) | 0 | 0 | 0 |
| SAP Replies(out) | 0 | 0 | 0 |
| SAP Discarded | 0 | 0 | 0 |

The elements of the display are described as follows:

SAP Statistics

| | |
|--------------------|---|
| SAP Updates (out) | Number of SAP updates generated by the router since the boot time or the last flush time. Periodic updates or incremental updates on serial interfaces are included in this category. Depending on the current number of SAP table entries, the number of SAP updates can vary. |
| SAP Updates (in) | Number of SAP broadcasts received on a port by the router. |
| SAP Requests (out) | Number of SAP requests generated by the router. Normally the router generates SAP requests on serial interfaces to learn new server information from the other router when a port comes up or new routes are learned from a remote router. |
| SAP Requests (in) | Number of SAP queries received by the router. |
| SAP Replies (out) | Number of SAP replies generated by the router in response to SAP queries. In general, the number of SAP replies is more than the number of SAP requests because most of the time more than one packet is required to satisfy one SAP request. |
| SAP Discarded | Number of SAP packets dropped by the router because of errors such as bad framed packets, and packets with unknown SAP packet types. |

SHDlc Service

The following is an example of the display generated by the sHow -SYS STATISTICS -SHDlc command:

```

ACCUMULATED VALUES
== SHDlc statistics ===== 3===== 3c=====
Frames:
Received                118767      25688
Transmitted             23758      128370
Bytes:
Received                48827832   51376
(cont'd)
ACCUMULATED VALUES(cont'd)
== SHDlc statistics ===== 3===== 3c=====
Transmitted             47516      13911937
Frames Discarded:
Received                16         44
Transmitted             0          5
Circuit Count:
Connected               61         2
Disconnected            0          0
    
```

The elements of the display are described as follows. Statistics correspond to the port numbers (3 and 3c) at the column heads.

Frames

Received Number of SDLC or HDLC frames received from the WAN by DLSw.
 Transmitted Number of SDLC or HDLC frames transmitted to the WAN by DLSw.

Bytes

Received Number of bytes received from the WAN by DLSw.
 Transmitted Number of bytes transmitted to the WAN by DLSw.

Frames Discarded

Received Number of SDLC or HDLC frames received from the WAN by DLSw that were discarded.
 Transmitted Number of SDLC or HDLC frames transmitted to the WAN by DLSw that were discarded

Circuit Count

Connected Number of times the circuit was in connection state.
 Disconnected Number of times the circuit was not in connected state.

SMDS Service

The following is an example of the display generated by the SHow -SYS STATISTICS -SMDS command:

```

ACCUMULATED VALUES
== SMDS statistics ===== 1===== 3===== 5===== 7=====
Packets Received:
  Individual Address   -         0         -         -
  Group Address        -         0         -         -
Packets Transmitted:
  Individual Address   -         0         -         -
  Group Address        -         0         -         -
Error Packets Received:
  Unrecognized IA      -         0         -         -
  Unrecognized GA      -         0         -         -
  Invalid Address Type -         0         -         -
    
```


Syntactic errors - 0 - -

The elements of this display are described as follows:

Packets Received

Individual Address Number of individually addressed SIP Level 3 PDUs received.
Group Address Number of group addressed SIP Level 3 PDUs received.

Packets Transmitted

Individual Address Number of individually addressed SIP Level 3 PDUs that have been sent out.
Group Address Number of group addressed SIP Level 3 PDUs that have been sent out.

Error Packets Received

Unrecognized IA Number of SIP Level 3 PDUs received with invalid or unknown individual destination.
Unrecognized GA Number of SIP Level 3 PDUs received with invalid or unknown group addresses.
Invalid Address Type Number of SIP Level 3 PDUs received that had the source or destination address_type fields (the four most significant bits of the address) not equal to the value 0xC or 0xE, or the value is equal to 0xE for the source address.
Syntactic errors Number of SIP Level 3 PDUs received that have errors, including protocol processing and bit errors, but excluding addressing related errors. For more information, see RFC 1304.

SNMP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -SNMP command:

```

ACCUMULATED VALUES
===== SNMP statistics =====
Incoming SNMP PDUs          39
  Get Requests               4
  Get-Next Requests         30
  Set Requests               0
  Bad PDUs                   5
Outgoing SNMP PDUs          34
  Get Responses              33
  Error Responses            1
  Traps                      0

```

The elements of this display are described as follows:

Incoming SNMP PDUs

Number of PDUs delivered to SNMP.

Get Requests Number of Get-Request PDUs processed by SNMP.
Get-Next Requests Number of Get-Next Request PDUs processed by SNMP.
Set Requests Number of Set-Request PDUs processed by SNMP.

Bad PDUs Number of PDUs delivered to but not processed by SNMP, including unsupported version, unknown community name, not allowed operation by the named community, ASN.1 parsing errors, and unknown PDU type.

Outgoing SNMP PDUs Number of PDUs generated by SNMP.

Get Responses Number of Get-Response PDUs generated by SNMP.
 Error Responses Number of valid SNMP PDUs generated by SNMP and for which the value of the error-status is not noError (0), including tooBig (1), noSuchName (2), badValue (3), readOnly (4), and genErr (5).
 Traps Number of Trap PDUs generated by SNMP.

SR Service

The following is an example of the display generated by the `sHow -SYS STATISTICS -SR` command:

```

ACCUMULATED VALUES
== SR statistics ===== 1===== 3===== 5===== 7=====
RECEIVED:
  All Route Explorer:      0         0         0         0
  Spanning Tree Explorer:  0         0         0         0
  Specifically Routed:     0         0         0         0
  SRT Gateway Packets     0         0         0         0
TRANSMITTED:
  All Route Explorer:      0         0         0         0
  Spanning Tree Explorer:  0         0         0         0
  Specifically Routed:     0         0         0         0
  SRT Gateway Packets     0         0         0         0
ERRORS:
  Bad Routing Info:       0         0         0         0
  Expl RD Limit Exceeded: 0         0         0         0
  "  Frames Too Long:    0         0         0         0
  "  Incorrect Ring In:  0         0         0         0
  SRF Duplicate Ring In:  0         0         0         0
  "  Missing Ring In:    0         0         0         0
  "  Bad Bridge Number:  0         0         0         0
  "  Bad Ring Out:       0         0         0         0
  Discarded SRTG Pkts:   0         0         0         0
  Unknown SRTG Pkts:     0         0         0         0
    
```

The elements of this display are described as follows:

RECEIVED

All Route Explorer: Number of All Route Explorer frames received.
 Spanning Tree Explorer: Number of Spanning Tree Explorer frames received.
 Specifically Routed: Number of Specifically Routed frames received.
 SRT Gateway Packets: Number or SRT gateway packets received.

TRANSMITTED

All Route Explorer: Number of All Route Explorer frames transmitted.

Spanning Tree Explorer: Number of Spanning Tree Explorer frames transmitted.
 Specifically Routed: Number of Specifically Routed frames transmitted.
 SRT Gateway Packets: Number of SRT gateway packets transmitted.

ERRORS

Bad Routing Info: Number of frames discarded because of a formatting error in the Routing Information field, for example, bad Largest Frame Size (LFS) or Direction bit (D) set in frame.

Expl RD Limit Exceeded: Number of explorer frames discarded because of the Routing Designator (RD) limit exceeded.

Expl Frames Too Long: Number of frames discarded because the size exceeds the largest frame size configured for the inbound interface.

Expl Incorrect Ring In: Number of explorer frames discarded because the last LAN ID of the RI does not equal the LAN-In ID.

SRF Duplicate Ring In: Number of Specifically Routed frames discarded because the LAN-In ID already existed in the RI.

SRF Missing Ring In: Number of Specifically Routed frames discarded because the LAN-In ID is not found in the RI field.

SRF Bad Bridge Number: Number of Specifically Routed frames discarded because the bridge number following the LAN-In ID does not match the bridge number of this bridge.

SRF Bad Ring Out: Number of Specifically Routed frames discarded because a matching LAN-Out ID (that is, the LAN ID that follows the bridge number of the local bridge) is not configured.

Discarded SRTG Pkts: Number of SRTG packets discarded due to incorrect packet formats.

Unknown SRTG Pkts: Number of packets discarded because their protocols are not supported.

STP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -STP command:

```

ACCUMULATED VALUES
== STP statistics ===== 1===== 3===== 5===== 7=====
STP statistics
Forwarding State Count      1          1          0          0
Blocking State Count        0          1          0          0
Bad Config BPDU received    12         1          0          0
Looped Config BPDU receiv   0          0          0          0
Message Age Timeouts        0          1          0          0
    
```

The elements of this display are described as follows:

STP statistics

Forwarding State Count: Number of times the specified port was put in forwarding state.

Blocking State Count: Number of times the specified port was put in blocking state because of a possible loop.

| | |
|-----------------------------|--|
| Bad Config BPDU received | Number of bridge protocol data units (BPDUs) received with information indicating that the transmitting bridge does not recognize another high-priority bridge on the network (possibly because the transmitting bridge just booted up). |
| Looped Config BPDU received | Number of BPDUs the bridge received its own. |
| Message Age Timeouts | Number of times a neighboring high-priority bridge has gone out of service. |

SYS Service

The following is an example of the display generated by the SHow -SYS DpmSTATistics POrt IP command:

```
DpmSTATistics --- From Source, Per Port, Protocol:IP
Destination ==>
Source
===      1 ===== 1B ===== 1C ===== 1D ===== 1E ===== 1F ===
1         0          0          0          592137      591940      591873
1B        0          0          0          592126      591833      591943
1C        0          0          0          592164      591735      591988
1D        0          0          0          0           0           0
1E        0          0          0          0           0           0
1F        0          0          0          0           0           0

===      4 ===== 5 ===== 7 ===== CEC ===
1         0          0          0          0
1B        0          0          0          0
1C        0          0          0          0
1D        0          0          0          11
1E        0          0          0          11
1F        0          0          0          11
```

The elements of the display are described as follows:

- Port** Per port statistics are displayed.
- Source** Data is displayed about packets transmitted from the specified slots or ports, for example, ports that are the source of the packets.

If the value is all zeroes for a particular source port, that port is not displayed.
- Protocol** Statistics for the specified protocol, IP, are displayed. In the screen example, 592164 IP packets were forwarded from port 1C to port 1D.

TCP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -TCP command:

```
ACCUMULATED VALUES
===== TCP statistics =====
TCP Packets:      Transmitted   Received      Retransmitted
                  728           245           0
TCP Connections:  Initiated     Accepted      Failed         Reset
```

0 2 0 0

The elements of this display are described as follows:

TCP Packets

| | |
|---------------|--|
| Transmitted | Number of TCP packets transmitted within a specified interval. |
| Received | Number of TCP packets received within a specified interval. |
| Retransmitted | Number of TCP packets retransmitted within a specified interval. Packets are retransmitted when the previous attempt fails or when they are timed out. |

TCP Connections

| | |
|-----------|---|
| Initiated | Number of TCP connections attempted. |
| Accepted | Number of successful TCP connection attempts. |
| Failed | Number of failed TCP connection attempts. |
| Reset | Number of TCP connections reset. Connections are reset if they were aborted because of error. |

UDP Service

The following is an example of the display generated by the `SHoW -SYS STATISTICS -UDP` command.



The UDP Service does not appear in the user interface. However, you can still obtain UDP statistics.

ACCUMULATED VALUES

===== UDP statistics =====

UDP Statistics:

| | |
|-----------------------------|-------|
| Datagrams transmitted | 323 |
| Good datagrams received | 83707 |
| Datagrams with errors | 0 |
| Datagrams with unknown port | 6 |
| ICMP Datagrams received | 0 |

The elements of this display are described as follows:

UDP Statistics

| | |
|-----------------------------|---|
| Datagrams transmitted | Number of packets sent to other networks. |
| Good datagrams received | Number of error-free packets received. |
| Datagrams with errors | Number of packets containing errors, such as checksum errors unable to give packet to UDP user. |
| Datagrams with unknown port | Number of packets received for which UDP does not have matching port number in its port table. |
| ICMP Datagrams received | Number of Internet Control Management Protocol packets received. |

UDPHELP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -UDPHELP command.



The system displays the UDPHELP statistics on a system-wide basis, rather than on a port-by-port basis.

```

ACCUMULATED VALUES
===== UDPHELP statistics =====
BOOTP/UDP/IP Broadcast Helper Statistics
  Packets Received           0
  Packets Forwarded          0
  Packets ReBroadcasted      0
  Packets Discarded           0
  Miscellaneous Error         0
    
```

The elements of this display are described as follows:

**BOOTP/UDP/IP
Broadcast Helper
Statistics**

- Packets Received Number of packets received.
- Packets Forwarded Number of packets forwarded to servers.
- Packets ReBroadcasted Number of packets forwarded to all interfaces.
- Packets Discarded Number of packets discarded because the packet did not broadcast a destination address.
- Miscellaneous Error Number of times a packet could not be forwarded because of a combination of heavy traffic on the network and memory constraints.

VIP Service

The following is an example of the display generated by the SHow -SYS STATISTICS -VIP command:

```

ACCUMULATED VALUES
== VIP statistics ===== 1===== 3===== 5===== 7=====
VINES IP Statistics:
Received           0         0         0         0
Xmitted            0         0         0         0
Forwarded          0         0         0         0
To Client          0         0         0         0
From Client        0         0         0         0
Discarded          0         0         0         0

VINES ARP Statistics:
Query Request(In)  0         0         0         0
Query Response(Out) 0         0         0         0
Assignment Req(In) 0         0         0         0
Assignment Resp(Out) 0         0         0         0
ARP Discarded      0         0         0         0

VINES ICP Statistics:
Exception          0         0         0         0
Metric             0         0         0         0

VINES RTP Statistics:
Updates(In)        0         0         0         0
Updates(Out)       0         0         0         0
Requests(In)       0         0         0         0
    
```

| | | | | |
|-------------------|---|---|---|---|
| Responses(Out) | 0 | 0 | 0 | 0 |
| Redirects(In) | 0 | 0 | 0 | 0 |
| Redirects(Out) | 0 | 0 | 0 | 0 |
| RTP Discarded(In) | 0 | 0 | 0 | 0 |
| Xmit Fail | 0 | 0 | 0 | 0 |

The elements of this display are described as follows:

VINES IP Statistics

| | |
|-------------|---|
| Received | Number of packets received on a port since boot-up time or last flushing. This number is the total number of packets received from the network including Forwarded packets, Broadcast and Unicast packets addressed to the router and successfully delivered to VIP clients, and some of Discarded packets. |
| Xmitted | Number of packets generated and transmitted by the router since boot-up time or last flushing. There can be only three types of packets (ARP, ICP, and RTP) generated by the router. |
| Forwarded | Number of packets routed successfully to other ports since boot-up time or last flushing. Those packets generated by the router itself are not included in this category. |
| To client | Number of broadcast packets or unicast packets addressed to the router and successfully delivered to proper VIP clients. ARP and ICP packets are not included in this count. So this number is the total number of RTP packets received on the port since boot-up time or last flushing. |
| From client | Number of broadcast packets or unicast packets received from the clients. On the router reside only VINES ARP, ICP, and RTP clients, but ARP and ICP packets are not included in this count; this number is the total number of RTP packets received on the port since boot-up time or last flushing. |
| Discarded | Number of packets discarded by VIP due to various errors such as bad framed packets, packets without any data, packets destined to other networks when VIP routing is turned off, packets addressed to the router but no clients available, etc. This number includes bad packets received from either networks or VIP clients. |

VINES ARP Statistics

| | |
|----------------------|--|
| Query Request(In) | Number of ARP Query Requests received. |
| Query Response(Out) | Number of ARP Query Responses generated. |
| Assignment Req(In) | Number of incoming ARP Assignment Requests. |
| Assignment Resp(Out) | Number of Assignment Responses generated. |
| ARP Discarded | Number of ARP packets discarded due to bad checksum, subnetwork number exhaustion, or various transmit failures. |

VINES ICP Statistics

| | |
|-----------|--|
| Exception | Number of exception notifications generated whenever bad packets were detected, but those packets have an error notify bit set by the source node. |
| Metric | Number of metric notifications generated whenever incoming packets have the metric bit set by the source node. |

VINES RTP Statistics

| | |
|-------------------|--|
| Updates(In) | Number of RTP broadcasts received on a port by the router. |
| Updates(Out) | Number of RTP broadcasts transmitted by the router since the boot time or last flushing. |
| Requests(In) | Number of RTP requests received by the router. |
| Replies(Out) | Number of RTP responses generated by the router in response to RTP requests. The number of RTP responses can be bigger than the number of RTP requests depending on the current number of RTP table entries. |
| Redirects(In) | Number of RTP Redirect packets received. |
| Redirects(Out) | Number of RTP Redirects generated. |
| RTP Discarded(In) | Number of RTP packets dropped by the router due to various errors such as packets received from unknown networks, lost packets, etc. |
| Xmit Failure(Out) | Number of instances where the router failed to generate RTP packets because of resource depletion. |

WE Service

The following is an example of the display generated by the SHow -SYS STATISTICS -WE command:

```

== WE statistics =====      2=====      3=====      6=====      8=====
Control Frames Sent           -             8199           -             -
Control Bytes Sent            -            106587         -             -
Control Frames Recv           -             8292           -             -
Control Bytes Recv            -            954097         -             -
LMI Frames Sent               -             8199           -             -
LMI Frames Recv               -             8199           -             -
Local Link Down Events        -              0              -             -

Received Frame Errors:
Invalid DLCI Frames           -              0              -             -
Invalid Control Frames        -              0              -             -
Inactive DLCI Discards        -              4              -             -
Small Frames Recv             -              0              -             -

```

The elements of this display are described as follows:

WE Statistics

| | |
|---------------------|---|
| Control Frames Sent | Specifies how many control frames were sent, using the WNI protocol, between the NETBuilder II bridge/router and the WAN Extender. Control frame counts include the LMI frame counts. |
| Control Bytes Sent | Specifies how many control bytes were sent, using the WNI protocol, between the NETBuilder II bridge/router and the WAN Extender. |
| Control Frames Recv | Specifies how many control frames were received, using the WNI protocol, between the NETBuilder II bridge/router and the WAN Extender. Control frame counts include the LMI frame counts. |

| | |
|------------------------|---|
| Control Bytes Received | Specifies how many control bytes were received, using the WNI protocol, between the NETBuilder II bridge/router and the WAN Extender. |
| LMI Frames Sent | Specifies how many LMI frames were sent, using the WNI protocol, between the NETBuilder II bridge/router and the WAN Extender. |
| LMI Frames Recv | Specifies how many LMI frames were received, using the WNI protocol, between the NETBuilder II bridge/router and the WAN Extender. |
| Local Link Down Events | Specifies the number of local link down events. These events occur when a number of consecutively unanswered Status Enquiry messages exceeds the ports configured ErrorThreshold. See the ErrorThreshold and KeepAliveInt parameters in <i>Reference for Enterprise OS Software</i> for more information about Status Enquiry messages. |

Received Frame Errors

| | |
|------------------------|---|
| Invalid DLCI Frames | Specifies how many DLCI frames sent between the NETBuilder II bridge/router and the WAN Extender were invalid. |
| Invalid Control Frames | Specifies how many control frames sent between the NETBuilder II bridge/router and the WAN Extender were invalid. |
| Inactive DLCI Discards | Specifies how many discarded data packets (because of a terminated connection) were encountered in transit after termination. |
| Small Frames Recv | Specifies how many small frames were received between the NETBuilder II bridge/router and the WAN Extender. |

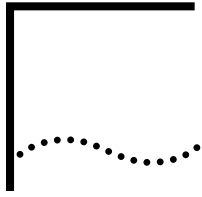
X25 Service

The following is an example of the display generated by the SHow [!<port>] -X25 X25STATistics command:

| | | | |
|--------------------|---|--------------------|---|
| Call Request | 0 | Incoming Call | 0 |
| Call Accepted | 0 | Call Connected | 0 |
| Clear Request | 0 | Clear Indication | 0 |
| DTE Clear Conf | 0 | DCE Clear Conf | 0 |
| DTE data | 0 | DCE Data | 0 |
| DTE Interrupt | 0 | DCE Interrupt | 0 |
| DTE Interrupt Conf | 0 | DCE Interrupt Conf | 0 |
| DTE RR | 0 | DCE RR | 0 |
| DTE RNR | 0 | DCE RNR | 0 |
| Reset Request | 0 | Reset Indication | 0 |
| DTE Reset Conf | 0 | DCE Reset Conf | 0 |
| Restart Request | 0 | Restart Indication | 0 |
| DTE Restart Conf | 0 | DCE Restart Conf | 0 |
| DTE Invalid Pkt | 0 | DCE Invalid Pkt | 0 |
| Link Down | 0 | Link Up | 0 |

The elements of this display are described as follows:

| | |
|--------------------|--|
| Call Request | Number of Call Request packets sent. |
| Call Accepted | Number of Calls Accepted packets sent. |
| Clear Request | Number of Clear Request packets sent. |
| DTE Clear Conf | Number of Clear Confirmation packets sent. |
| DTE data | Number of Data packets sent. |
| DTE Interrupt | Number of Interrupt Request packets sent. |
| DTE Interrupt Conf | Number of Interrupt Confirmation packets sent. |
| DTE RR | Number of RR packets sent. |
| DTE RNR | Number of RNR packets sent. |
| Reset Request | Number of Reset Request packets sent. |
| DTE Reset Conf | Number of Reset Confirmation packets sent. |
| Restart Request | Number of Restart Request packets sent. |
| DTE Restart Conf | Number of Restart Confirmation packets sent. |
| DTE Invalid Pkt | Number of Invalid Packets received from clients. |
| Link Down | Number of times Frame Layer went down. |
| Incoming Call | Number of Incoming calls received. |
| Call Connected | Number of Call Connected packets received. |
| Clear Indication | Number of Clear Indication packets received. |
| DCE CLeat Conf | Number of Clear Confirmation packets received. |
| DCE Data | Number of Data packets received. |
| DCE Interrupt | Number of Interrupt packets received. |
| DCE Interrupt Conf | Number of Interrupt Confirmation packets received. |
| DCE RR | Number of RRs received. |
| DCE RNR | Number of RNRs received. |
| Reset Indication | Number of Reset Indication packets received. |
| DCE Reset Conf | Number of Reset Confirmation packets received. |
| Restart Indication | Number of Restart Indication packets received. |
| DCE Restart Conf | Number of Restart Confirmation packets received. |
| DCE Invalid Pkt | Number of Unrecognized packets received. |
| Link Up | Number of times Frame Layer came up. |



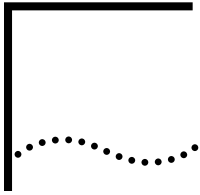
STATIC TABLES

The number of entries you can have in statically configured routing tables depends on the NETBuilder hardware on which you are running your bridge/router software.

Table 128 lists the different bridge/router features and router types, and the maximum number of routing table entries you can have for each hardware platform.

Table 128 Number of Entries Allowed in Static Tables

| Bridge/Router Feature | NB II (all chassis) | SS II 222, 227, and 228 | SS II 422 and 427 |
|------------------------------|----------------------------|--------------------------------|--------------------------|
| Bridge | | | |
| Bridge Table | 2048 | 512 | 512 |
| AppleTalk | | | |
| Address Mapping Table | 1000 | 1000 | 1000 |
| Banyan VINES | | | |
| WAN Neighbor Table | 128 | 64 | 64 |
| DECnet | | | |
| WAN Neighbor Table | 32/WAN port | 32/WAN port | 32/WAN port |
| IP | | | |
| Static Routing Table | 256 | No limit | No limit |
| Static Address Table | 256 | No limit | No limit |
| Secondary IP Addresses | 32 | No limit | No limit |
| IP-OSPF | | | |
| ExteriorPolicy | 64 | No limit | No limit |
| InteriorPolicy | 64 | No limit | No limit |
| StaticPolicy | 64 | No limit | No limit |
| ReceivePolicy | 64 | No limit | No limit |
| Neighbors | 16/port | No limit | No limit |
| Virtual Link | 8/port | No limit | No limit |
| OSI | | | |
| End System Table | 64 | 64 | 64 |
| PrefixRoutes | 64 | 64 | 64 |
| Neighbors | 28 | 28 | 28 |



AUDIT TRAIL MESSAGES

Table 129 describes bridge/router audit trail messages identified by record type and status codes. For example, a message that indicates a macro cache overflow includes the record type code MO.

Table 129 Audit Trail Messages

| Service | Record Type Code | Status Code | Description |
|---------------------|------------------|-------------|---|
| All services | CC | | The user has configured the network. Each CC record type is followed by an additional explanatory code, which indicates what commands or parameters have been used. The record also usually displays changed values. |
| ARP | AC | | Duplicate Internet address detected. |
| BRidge | | FW | The CONTRol parameter has been set to FOrward or NoFOrward as indicated in the record. |
| | | LE | The CONTRol parameter has been set to LEarn or NoLEarn as indicated in the record. |
| | | SRS FWD | The SRcSecurity parameter has been set to Fwd for the specified port. |
| | | SRS BLK | The SRcSecurity parameter has been set to Blk for the specified port. |
| | | SRS NONE | The SRcSecurity parameter has been set to None for the specified port. |
| | | DSS FWD | The DStSecurity parameter has been set to Fwd for the specified port. |
| | | DSS BLK | The DStSecurity parameter has been set to Blk for the specified port. |
| | | DSS NONE | The DStSecurity parameter has been set to None for the specified port. |
| IP | IR | | An ICMP message from the second device has been received by the first device indicated on the audit trail. The two numbers that follow are the type field and code field of the ICMP message. Type 3 messages (known as destination unreachable) contain additional information from the returned (erroneous) IP header. For more information, see RFC 792. |
| | IX | | An ICMP message has been transmitted from the first device to the second device indicated on the audit trail. The two numbers that follow are the type field and code field of this ICMP message. |
| Macro | MO | | Macro cache overflow. Macro service may be disrupted for some users. |
| | MI | | Message generated by a macro currently in execution. AUDIT <message> is inside a macro. <Message> can be anything. |
| PATH | CC | PA ON | The CONTRol parameter has been used to enable the specified path. |
| | | PA OFF | The CONTRol parameter has been used to disable the specified path. |
| | | BA | The BAud parameter has been used to set the baud rate for the specified path to the value indicated in the record. |
| | NU | | Indicates the average per-minute utilization of network capacity (on a scale of 1,000). This record is generated once every 10 minutes. Each record consists of 10 entries, one for each minute. To obtain the percentage of network capacity represented by an entry, divide the reported number by 10. |
| | PU | | The specified path is operating. |
| PATH | SUX | | Records the average per-minute outgoing traffic as a fraction of the specified serial line capacity (on a scale of 1,000). This record is generated once every 10 minutes. Each record consists of 10 entries, one for each minute. To obtain the percentage of outgoing capacity represented by an entry, divide the reported number by 10. |

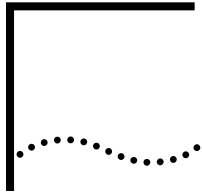
Table 129 Audit Trail Messages (continued)

| Service | Record Type Code | Status Code | Description | |
|----------------------|------------------|-------------|---|---|
| | SUR | | Records the average per-minute incoming traffic as a percentage of the specified serial line capacity (on a scale of 1,000). This record is generated once every 10 minutes. Each record consists of 10 entries, one for each minute. To obtain the percentage of incoming capacity represented by an entry, divide the reported number by 10. | |
| PORT | CC | PORT ON | The specified port has been enabled. | |
| | CC | PORT OFF | The specified port has been disabled. | |
| | CC | ADD PA | Paths have been added to the specified port. | |
| | CC | DEL PA | Paths have been deleted from the specified port. | |
| PPP | AC | | Duplicate Internet address detected. | |
| Serial Line Protocol | PD | NOACK | The specified path is not operating because acknowledgments for probe packets are not being received. | |
| SLIP | AC | | Duplicate Internet address detected. | |
| STP | BC | | The specified bridge has been booted. | |
| | CC | STPPRI | The PortPriority parameter has been set to the value indicated. | |
| | CC | STPPRI | The BridgePriority parameter has been set to the value indicated. | |
| | CC | STPCOST | The PathCost parameter has been set to the value indicated. | |
| | BK | | The specified port is in blocking mode. | |
| | FW | | The specified port is in forwarding mode. | |
| | CC | STSTP ON | The Spanning Tree Protocol has been turned on. | |
| | CC | STSTP OFF | The Spanning Tree Protocol has been turned off. | |
| | CC | STHR ON | HopReduce has been selected for the Spanning Tree Protocol. | |
| | CC | STHR OFF | NoHopReduce has been selected for the Spanning Tree Protocol. | |
| SYS | None | None | <title of statistics> exceeded per minute threshold <threshold>
<title of statistics> exceeded per hour threshold <threshold>
<title of statistics> exceeded per day threshold <threshold>
<title of statistics> exceeded accumulation threshold

These messages indicate that normal levels of network activity have been exceeded for a specific statistic. No record type code or status code appears in the message. | |
| | CD | | A connection was established between the specified devices using the indicated protocol. | |
| | CF | | An attempt to establish a connection between the specified devices failed. | |
| | DC | | The connection between the specified devices was disconnected. | |
| | DLT | SN | | A data link test was initiated from the local bridge to a wide area bridge. The record indicates whether the test was run at the maximum transmission rate, which is the default transmission rate, or at a user-defined transmission rate. The record also contains the address of the wide area bridge. |
| | | ST | | A data link test was initiated from a wide area bridge to a local bridge. The record also contains the address of the wide area bridge. |
| | | DONE | | A data link test has been completed. |
| | LS | | | The specified device entered Listening mode. |
| | NLI | | | This message indicates that the subject of a network login request sent a report to the NCS. |
| | | | OK | Indicates that the status of the login request is that the keys certified and the request was granted. |
| | | | NR | No response to the login request. |

Table 129 Audit Trail Messages (continued)

| Service | Record Type Code | Status Code | Description |
|----------------|-------------------------|--------------------|---|
| | | SE | System error. |
| | | UN | Denied request for User Profile because the profile was not found. |
| | | PE | Password error. |
| | | DN | Denied request for User Profile because the request was not allowed. |
| PCS | | | This message indicates tha the subject of a password change sent a report to the NCS. |
| | | NR | No response to the password change request. |
| | | SE | System error. |
| | | DN | The request for User Profile was denied because the request was not allowed. |
| | | OK | Indicates that the password change request key was certified and the request granted. |
| | PE | | Five unsuccessful attempts to enter a password on the internetwork bridge have been made during the five minutes before the message was recorded. |
| | UPL | | This message indicates that the network user changed the privilege at the specified port |
| | | OK | Indicates that the keys certified and the request was granted. |
| | | PE | Indicates that there was a password error |



SYSLOG MESSAGES

The following table describes bridge/router syslog messages identified by record type and status codes.



Italicized items in the log string are substituted with actual values.

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|----------|---|----------------------------------|---|-----------|-----------------------------------|
| AuditLog | 100
101
102
103
104
105
106
107
108 | Auditlog file creation failure | " AuditLog log file not created - <i>reason</i> "
reason can be:
- bad pdmp size from DUMP
- fail memory allocation for DUMP
- file write error
- media not present
- media write protected
- directory is full
- unsupported media
- file system error
- missing signature | LogError | AuditLog service control & config |
| AuditLog | 109 | Auditlog file read failure | " AuditLog Read logfile or Read auditlog.dmp error" | LogError | AuditLog service control & config |
| AuditLog | 110 | Auditlog file access status | " AuditLog no audlog or no log file error" | LogError | AuditLog service control & config |
| AuditLog | 111 | Auditlog file created successful | " AuditLog log file %s/%s created" | LogNotice | AuditLog service control & config |
| AuditLog | 112 | Auditlog file logging disabled | " AuditLog file logging disabled - MaxNumLogFile = 0" | LogNotice | AuditLog service control & config |
| DHCP | 300 | DHCP address pool out of address | " dhcp_discover out of addresses" | LogAlert | AuditLog service control & config |
| DHCP | 301 | DHCP address offering | " OFFERING IP(<i>ip_addr</i>) to client(<i>client_id</i>)" | LogAlert | AuditLog service control & config |
| DHCP | 302 | DHCP address released | " receives DHCPRELEASE and release IP(<i>ip_addr</i>) from client(<i>client_id</i>)" | LogAlert | AuditLog service control & config |
| DHCP | 303 | DHCP RAS address allocation | " dhcp_ras_getip - Got IP address (<i>ip_addr</i>)" | LogAlert | AuditLog service control & config |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|----------------------------|--|----------|-----------------------------------|
| System | 400 | Bandwidth status | " BW TARGET MET FOR PORT <i>port_num</i> : NORMAL BAND WIDTH IS ZERO" | LogInfo | AuditLog service control & config |
| System | 401 | Bandwidth status | " BW TARGET NOT MET FOR PORT <i>port_num</i> : <i>reason</i> "
reason can be:
DIAL POOL EMPTY/USED
DIAL NO. LIST EMPTY
NO MATCH FOUND
NO MORE MEMORY
PORT NOT DIALABLE
NO MORE PATH/DIAL NO.
PORT MIS-CONFIGURED
REASON UNKNOWN | LogInfo | AuditLog service control & config |
| System | 402 | Dial-on-Demand port status | " DOD PORT <i>port_num</i> DOWN - NO PATH(S) AVAILABLE" | LogInfo | AuditLog service control & config |
| System | 403 | Outgoing call failure | " RETRY COUNT ON PORT <i>port_num</i> EXCEEDED" | LogInfo | AuditLog service control & config |
| System | 404 | Call Termination status | " INITIATING HANGUP ON PATH <i>path_num</i> " | LogInfo | AuditLog service control & config |
| System | 405 | Outgoing call failure | " CALL ON PATH <i>path_num</i> REJECTED, NO CARRIER" | LogInfo | AuditLog service control & config |
| System | 406 | Outgoing call failure | " CALL ON PATH <i>path_num</i> REJECTED, PATH DID NOT COME UP" | LogInfo | AuditLog service control & config |
| System | 407 | Incoming call status | " INCOMING CALL ON DIALPOOL PATH <i>path_num</i> " | LogInfo | AuditLog service control & config |
| System | 408 | Incoming call failure | " INCOMING CALL ON DIALPOOL PATH <i>path_num</i> REJECTED" | LogInfo | AuditLog service control & config |
| System | 409 | Incoming call failure | " INCOMING CALL ON PATH <i>path_num</i> REJECTED. NO PORT." | LogInfo | AuditLog service control & config |
| System | 410 | Incoming call success | " INCOMING CALL ON DIALPOOL PATH <i>path_num</i> CONNECTED" | LogInfo | AuditLog service control & config |
| System | 411 | Incoming call success | " INCOMING CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> CONECTED" | LogInfo | AuditLog service control & config |
| System | 412 | Call termination status | " DISCONNECT ON PATH <i>path_num</i> " | LogInfo | AuditLog service control & config |
| System | 413 | Incoming call failure | " CALLER ID MISSING ON DIAL PATH <i>path_num</i> ." | LogInfo | AuditLog service control & config |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|-------------------------|---|----------|-----------------------------------|
| System | 414 | Call termination status | <i>hangup_initiator</i> INITIATING HANGUP ON PATH <i>path_num</i> , PORT <i>port_num</i>

<i>hangup_initiator</i> can be:
USER,
BOD,
DOD,
PATH NOT COMING UP,
PATH IS IDLE | LogInfo | AuditLog service control & config |
| System | 415 | Call termination status | <i>hangup_initiator</i> INITIATING HANGUP ON PATH

<i>hangup_initiator</i> can be:
USER,
BOD,
DOD,
PATH NOT COMING UP,
PATH IS IDLE | LogInfo | AuditLog service control & config |
| System | 416 | Incoming call failure | " CALLER ID <i>caller_id</i> ON PATH <i>path_num</i> NOT FOUND." | LogInfo | AuditLog service control & config |
| System | 417 | Outgoing call status | " INITIATING CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| System | 418 | Call termination status | " INITIATING HANGUP ON PATH <i>path_num</i> , PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| System | 419 | Outgoing call failure | " CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> REJECTED, NO CARRIER" | LogInfo | AuditLog service control & config |
| System | 420 | Call failure | " CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> REJECTED, PATH DID NOT COME UP" | LogInfo | AuditLog service control & config |
| System | 421 | Incoming call status | " INCOMING CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| System | 422 | Incoming call failure | " INCOMING CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> REJECTED" | LogInfo | AuditLog service control & config |
| System | 423 | Outgoing call failure | " CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> REJECTED, PREVIOUS HANGUP NOT YET COMPLETED" | LogInfo | AuditLog service control & config |
| System | 424 | Outgoing call failure | " CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> REJECTED" | LogInfo | AuditLog service control & config |
| System | 425 | Outgoing call status | " OUTGOING CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> CONNECTED" | LogInfo | AuditLog service control & config |
| System | 426 | Incoming call status | " INCOMING CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> CONNECTED" | LogInfo | AuditLog service control & config |
| System | 427 | Call termination status | " DISCONNECT ON PATH <i>path_num</i> , PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|-----------------------|---|----------|-----------------------------------|
| System | 428 | Incoming call failure | " INCOMING CALL ON PATH <i>path_num</i> REJECTED; PORT <i>port_num</i> DISABLED" | LogInfo | AuditLog service control & config |
| System | 429 | Incoming call failure | " INCOMING CALL ON PATH <i>path_num</i> REJECTED; PORT <i>port_num</i> OWNER IS NOT PPP" | LogInfo | AuditLog service control & config |
| System | 430 | Incoming call failure | " INCOMING CALL ON PATH <i>path_num</i> REJECTED BY PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| System | 431 | Incoming call failure | " INCOMING CALL ON PATH <i>path_num</i> REJECTED; PATH NOT IN PORT <i>port_num</i> POOL PREF LIST" | LogInfo | AuditLog service control & config |
| System | 432 | Incoming call status | " INCOMING CALL ON PATH <i>path_num</i> BOUND TO PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| System | 433 | Leased path status | " DYNAMIC LEASED PATH <i>path_num</i> BOUND TO PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| System | 434 | Outgoing call failure | " ISDN CALL ON PATH % <i>path_num</i> %, PORT % <i>port_num</i> % REJECTED, CAUSE CODE = % <i>code_num</i> %" | LogInfo | AuditLog service control & config |
| System | 435 | Outgoing call failure | " ISDN CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> REJECTED, CAUSE CODE = <i>code_num</i> (<i>code_string</i>)"
<i>code_string</i> can be:
" NORMAL CALL CLEARING"
" USER BUSY"
" NO USER RESPONDING"
" CALL REJECTED"
" DESTINATION OUT OF ORDER"
" NO CIRCUIT AVAILABLE"
" NETWORK OUT OF ORDER"
" TEMPORARY FAILURE"
" NETWORK CONGESTION"
" INVALID INFO. ELEMENT CONTENTS"
" MESSAGE INCOMPATIBLE WITH CALL STATE"
" PREVIOUS HANGUP NOT YET COMPLETED" | LogInfo | AuditLog service control & config |
| System | 436 | Outgoing call failure | " CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> REJECTED, CODE = <i>reject_code</i> " | LogInfo | AuditLog service control & config |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|--|---|----------|-----------------------------------|
| System | 437 | Outgoing call status | <i>initiator</i> INITIATING CALL ON PATH <i>path_num</i> , PORT <i>port_num</i>
initiator can be:
" PATH NOT COMING UP,"
" PATH IS IDLE,"
" USER"
" DOD"
" BOD" | LogInfo | AuditLog service control & config |
| System | 438 | Incoming call failure (Caller Line Identification) | " CLI SECURITY VIOLATION ON PATH <i>path_num</i> , PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| System | 439 | Incoming call failure (Caller Line Identification) | " CLI IS MISSING IN INCOMING CALL ON PATH <i>path_num</i> , PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| System | 440 | Incoming call failure (Caller Line Identification) | " CLI <i>cli_code</i> SECURITY VIOLATION ON PATH <i>path_num</i> , PORT <i>port_num</i> " | LogInfo | AuditLog service control & config |
| PORT | 441 | | EXPECTING CALL BACK ON PORT %port_num% | LogInfo | AuditLog service control & config |
| PORT | 442 | | DOD RESET FAILED ON PORT port_num PATH %path_num%: %reason% reason is: DIAL NO. LIST EMPTY NO MATCH DIAL NO. NO MORE MEMORY | LogInfo | AuditLog service control & config |
| PORT | 443 | | ISDN CALL ON PATH <i>path_num</i> , PORT %port_num% REJECTED, DUE TO CALL BACK | LogInfo | AuditLog service control & config |
| PORT | 444 | | CALL BACK MATCH CLI <i>cli_code</i> ON PATH %path_num% , PORT %port_num% | LogInfo | AuditLog service control & config |
| PORT | 445 | | CALL BACK TIME OUT EXPIRED ON PORT %port_num% | LogInfo | AuditLog service control & config |
| PORT | 446 | | CALL BACK TIME OUT CANCELLED ON PORT %port_num% | LogInfo | AuditLog service control & config |
| PORT | 447 | | bm_bring_down_bw: BM MEMORY ALLOCATION ERROR | LogInfo | AuditLog service control & config |
| PORT | 448 | | BM PANIC -- %text% | LogInfo | AuditLog service control & config |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|----------|--------|---|---|--|----------------------------|
| FireWall | 501 | Firewall action log
(protocol = TCP) | Tx/Rx ! <i>port_label</i>
src_ip_addr(<i>src_port_name</i>)
->dest_ip_addr(<i>dest_port_name</i>)
TCP, Permit/Deny, <i>info</i>

info can be:
TinyFragment,
SrcSpoof,
IPTunnel,
InFilter,
DefActionIn,
SrcRoute,
RecordRoute,
TimeStamp,
OutFilter,
DefActionOut | User configured via firewall setup, default is LogINfo | firewall log set to SysLog |
| FireWall | 502 | Firewall action log
(protocol == ICMP) | Tx/Rx ! <i>port_label</i> src_ip_addr
->dest_ip_addr ICMP(<i>icmp_type</i>),
Permit/Deny, <i>info</i>

icmp_type can be:
DestUnreach
SrcQuench
ReDirect
EchoReq
EchoRsp
TimeExcd
ParamProblem
DestAdmUnreach
info can be:
TinyFragment,
SrcSpoof,
IPTunnel,
InFilter,
DefActionIn,
SrcRoute,
RecordRoute,
TimeStamp,
OutFilter,
DefActionOut | User configured via firewall setup, default is LogINfo | firewall log set to SysLog |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|----------|--------|--|---|--|--------------------------------------|
| FireWall | 503 | Firewall action log
(protocol is not TCP)
and (protocol is not ICMP) | Tx/Rx ! <i>port_label</i> <i>src_ip_addr</i>
-> <i>dest_ip_addr</i> <i>protocol</i> , Permit/Deny,
info
protocol can be:
UDP, OSPF, ESP, AH, GRE,
ALL_PROTOCOLS
info can be:
TinyFragment,
SrcSpoof,
IPTunnel,
InFilter,
DefActionIn,
SrcRoute,
RecordRoute,
TimeStamp,
OutFilter,
DefActionOut | User
configured
via firewall
setup,
default is
LogInfo | firewall log set to
SysLog |
| IPSEC | 600 | IPSEC Log Overflow | " IPSEC: <i>num_log_msg</i> log messages
suppressed" | LogInfo | IPSEC LogDest set
to SysLog |
| IPSEC | 601 | IPSEC negotiation
failure | " <i>phase</i> SA Negotiation with <i>ip_addr</i>
failed: <i>fail_reason</i> " | LogInfo | IPSEC LogDest set
to SysLog |
| IPSEC | 649 | | Test message number <i>test_num</i> | LogInfo | IPSEC LogDest set
to SysLog |
| IPSEC | 650 | KEK configured | " Key Encryption Key Configured" | LogWarning | IPSEC LogDest set
to SysLog |
| IPSEC | 651 | KEK reset | " Key Encryption Key Reset" | LogWarning | IPSEC LogDest set
to SysLog |
| IPSEC | 652 | KEK secured
information
removed | " <i>user</i> encrypted configuration
information purged." | LogWarning | AuditLog service
control & config |
| NAT | 800 | NAT log overflow
(when log rate is
over 10 per second) | <i>num_msg</i> previously unreported log
messages, due to overflow | LogInfo | NAT log set to
Syslog |
| NAT | 801 | NAT Session
Successful for
TCP/UDP | Inbound/Outbound ! <i>port_label</i> TCP/UDP
(<i>src_ip</i> , <i>src_tuport</i>)
->(<i>dest_ip</i> , <i>dest_tuport</i>), translated to
<i>pkt_direction</i> (<i>xlate_ip</i> , <i>xlate_port</i>)
<i>pkt_direction</i> can be:
" be destined to" ,
" originate from" | configured
by user
using the
" TcpUdpPor
tMap"
parameter
under the
NAT service
(default is
LogInfo) | NAT log set to
Syslog |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|--|---|---|-----------------------|
| NAT | 802 | NAT Session Failure for TCP/UDP | <p>Inbound/Outbound !<i>port_label</i>/TCP/UDP (<i>src_ip,src_tuport</i>)
 ->(<i>dest_ip,dest_tuport</i>), translation failed
 - <i>failure_reason</i></p> <p>failure_reason can be:
 " No matching MAP found." ,
 " TUport not found in free tuport list." ,
 " " Packet direction is invalid." ,
 IP protocol not permitted." ,
 " Ran out of NAT addresses." ,
 " Ran out of TCP/UDP ports." ,
 " Ran out of Load share hosts." ,
 " No matching NAT address." ,
 " Could be due to premature session time-out." ,
 " Cannot create any more sessions." ,
 " Too many retries without a response." ,
 " Packet data length is too small."</p> | configured by user using the "TcpUdpPortMap" parameter under the NAT service (default is LogINfo) | NAT log set to Syslog |
| NAT | 803 | NAT Session successful for protocol = ICMP | <p>Inbound/Outbound !<i>port_label</i>/ICMP(<i>icmp_type</i>), <i>src_ip</i>
 -><i>dest_ip</i> translated to <i>pkt_direction</i>
 <i>xlate_ip</i></p> <p>icmp_type can be:
 DestUnreach
 SrcQuench
 ReDirect
 EchoReq
 EchoRsp
 TimeExcd
 ParamProblem
 DestAdmUnreach</p> <p>pkt_direction can be:
 " be destined to" ,
 " originate from"</p> | configured by user using the "TcpUdpPortMap" parameter under the NAT service (default is LogINfo) | NAT log set to Syslog |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|---|---|---|-----------------------|
| NAT | 804 | NAT Session Failure for protocol = ICMP | <p>Inbound/Outbound !<i>port_label</i>
ICMP(<i>icmp_type</i>), <i>src_ip</i>
-><i>dest_ip</i> translation failed: <i>fail_reason</i></p> <p>icmp_type can be:
DestUnreach
SrcQuench
ReDirect
EchoReq
EchoRsp
TimeExcd
ParamProblem
DestAdmUnreach</p> <p>fail_reason can be:
" No matching MAP found." ,
" TUport not found in free tuport list." ,
" IP protocol not permitted." ,
" Packet direction is invalid." ,
" Ran out of NAT addresses." ,
" Ran out of TCP/UDP ports." ,
" Ran out of Load share hosts." ,
" Ran out of fragmem." ,
" No matching NAT address." ,
" Could be due to premature session time-out." ,
" Cannot create any more sessions." ,
" Too many retries without a response." ,
" Cannot get internal buffer." ,
" Packet data length is too small."</p> | configured by user using the "TcpUdpPortMap" parameter under the NAT service (default is LogInfo) | NAT log set to Syslog |
| NAT | 805 | NAT Session Successful for protocol is not one of the following: (TCP UDP ICMP) | <p>Inbound/Outbound !<i>port_label</i>
<i>protocol_num</i> <i>src_ip</i>-><i>dest_ip</i> translated to <i>pkt_direction</i> <i>xlate_ip</i></p> <p>pkt_direction can be:
"be destined to" ,
"originate from"</p> | configured by user using the "TcpUdpPortMap" parameter under the NAT service (default is LogInfo) | NAT log set to Syslog |

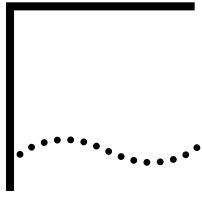
| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|--|--|--|--|
| NAT | 806 | NAT Session Failure for protocol != (TCP UDP ICMP) | Inbound/Outbound <i>!port_label protocol src_ip->dest_ip</i> translation failed:
<i>fail_reason</i>
fail_reason can be:
" No matching MAP found." ,
" TUport not found in free tuport list." ,
" IP protocol not permitted." ,
" Packet direction is invalid." ,
" Ran out of NAT addresses." ,
" Ran out of TCP/UDP ports." ,
" Ran out of Load share hosts." ,
" Ran out of fragmem." ,
" No matching NAT address." ,
" Could be due to premature session time-out." ,
" Cannot create any more sessions." ,
" Too many retries without a response." ,
" Cannot get internal buffer." ,
" Packet data length is too small." | configured by user using the "TcpUdpPort Map" parameter under the NAT service (default is LogInfo) | NAT log set to Syslog |
| RAS | 900 | RAS LogInSuccess
(with no tunnel IP address) | User <i>username</i> Logged In On Port <i>port_num</i> , Path <i>path_num</i> | LogAlert | RAS Log parameter set to SysLog and auditLog server ip |
| RAS | 901 | RAS LogInSuccess
(with tunnel IP address) | User <i>username</i> Logged In On Port <i>port_num</i> , Path <i>path_num</i> - Tunnel Remote IP Addr <i>IP_address</i> | LogAlert | RAS Log parameter set to SysLog and auditLog server ip |
| RAS | 902 | RAS LogOut | User <i>username</i> Idled/Logged Out of Port <i>port_num</i> , path <i>path_num</i> - IP address <i>IP_address</i> - Tunnel Remote IP Addr <i>IP_address</i> - Session start time <i>time_string</i> - Duration <i>duration_time</i> - Pkts In <i>num_pkts</i> Out <i>num_pkts</i> - Bytes In <i>num_bytes</i> Out <i>num_bytes</i> | LogInfo | RAS Log parameter set to SysLog and auditLog server ip |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------------|---|--|------------|--|
| RAS | 903 | RAS LogInFailure | <i>username</i> Login Failed on Path <i>path_num</i> : <i>reason</i>
Reason can be:
Authorization Failure
UserID Exceeds Max Length
UserID Invalid
Password Exceeds Max Length
Password Invalid
Challenge Response Invalid
RAS Disabled
No Port Available | LogAlert | RAS Log parameter set to SysLog and auditLog server ip |
| RAS | 904 | RAS ConnFailure | <LCP NCP> Failure on Path <i>path_num</i> , User <i>username</i>
Note: either LCP or NCP is specified | LogNotice | RAS Log parameter set to SysLog and auditLog server ip |
| RAS | 905 | RAS PathLogInSuccess
(with no tunnel IP address) | User <i>username</i> Logged In On TunnelSwitch Port <i>port_num</i> , Path <i>path_num</i> | LogAlert | RAS Log parameter set to SysLog and auditLog server ip |
| RAS | 906 | RAS PathLogInSuccess
(with tunnel IP address) | User <i>username</i> Logged In On TunnelSwitch Port <i>port_num</i> , Path <i>path_num</i> - Tunnel Remote IP Address <i>IP_address</i> | LogAlert | RAS Log parameter set to SysLog and auditLog server ip |
| RAS | 907 | RAS PathLogOut | User <i>username</i> Idled/Logged Out Of TunnelSwitch Port <i>port_num</i> , Path <i>path_num</i> - Pkts In <i>num_pkts</i> Out <i>num_pkts</i> - Bytes In <i>num_bytes</i> Out <i>num_bytes</i> | LogInfo | RAS Log parameter set to SysLog and auditLog server ip |
| REMP | 1000 to 1014 | RemPolling poll result | Encoded remote polling results, check REMPPolling doc | LogInfo | AuditLog service control & config |
| SNMP | 1200 | SNMP set requests | "SNMP <i>snmp_oid value</i> " for snmp SET requests
Example:
SNMP 129.213.144.69
"1.3.6.1.2.1.1.5.0
119.121.99.95.100.112.101" | LogWarning | AuditLog service control & config |
| SNMP | 1201 | | "% <i>snmp_oid_value</i> %" for SNMP SET requests with invalid community string | LogAlert | AuditLog service control & config |
| System | 1300 | Console login failure | "Login Failed" | LogAlert | AuditLog service control & config |
| System | 1301 | Console login success | "Login Successful" | LogAlert | AuditLog service control & config |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|----------------------------------|---|------------|-----------------------------------|
| System | 1302 | Console session termination | " UI session terminated" | LogAlert | AuditLog service control & config |
| System | 1303 | Console listen command | " Listen executed" | LogAlert | AuditLog service control & config |
| System | 1304 | Console privilege change failure | " Set Privilege Failed" | LogAlert | AuditLog service control & config |
| System | 1305 | Reboot command executed | " monitor reboot" | LogNotice | AuditLog service control & config |
| System | 1306 | System startup | " System Booted" | LogNotice | AuditLog service control & config |
| System | 1307 | Loop detected | " Loop Detected" | LogNotice | AuditLog service control & config |
| System | 1308 | System initialized | " System Initialized and Running" | LogNotice | AuditLog service control & config |
| System | 1309 | Path up | " Path <i>path_num</i> Up" | LogNotice | AuditLog service control & config |
| System | 1310 | Path down | " Path <i>path_num</i> Down" | LogNotice | AuditLog service control & config |
| System | 1311 | Path faulty | " Path <i>path_num</i> Faulty" | LogNotice | AuditLog service control & config |
| System | 1312 | path loopback | " Path <i>path_num</i> Loopback" | LogNotice | AuditLog service control & config |
| System | 1313 | | System restart due to: %text% | LogAlert | AuditLog service control & config |
| System | 1314 | | System restart without partial dump | LogAlert | AuditLog service control & config |
| System | 1315 | | Invalid Command: %cmd_text% | LogWarning | AuditLog service control & config |
| System | 1316 | | Write to disk of New Password for %user% failed | LogAlert | AuditLog service control & config |
| System | 1317 | | Failed adding user : '<unknown user name>' | LogAlert | AuditLog service control & config |
| System | 1318 | | Failed adding '%user%' | LogAlert | AuditLog service control & config |
| System | 1319 | | User '%user%' is added | LogAlert | AuditLog service control & config |
| System | 1320 | | Failed adding user '%user%' : name duplicated | LogAlert | AuditLog service control & config |
| System | 1321 | | Failed adding user '%user%' | LogAlert | AuditLog service control & config |
| System | 1322 | | Failed deleting user : <unknown user name> | LogAlert | AuditLog service control & config |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|-------------------------|---|-----------|-----------------------------------|
| System | 1323 | | Failed deleting user '%user%' | LogALert | AuditLog service control & config |
| System | 1324 | | User '%user%' is deleted | LogALert | AuditLog service control & config |
| System | 1325 | | Failed changing password : <unknown user name> | LogALert | AuditLog service control & config |
| System | 1326 | | Failed changing password : '%user%' | LogALert | AuditLog service control & config |
| System | 1327 | | Password of '%user%' has been changed by user | LogALert | AuditLog service control & config |
| System | 1328 | | Failed changing password for '%user%' | LogALert | AuditLog service control & config |
| System | 1329 | | Failed expiring user : <unknown user name> | LogALert | AuditLog service control & config |
| System | 1330 | | Failed expiring '%user%' | LogALert | AuditLog service control & config |
| System | 1331 | | User '%user%' has been expired | LogALert | AuditLog service control & config |
| System | 1350 | General system messages | <p><i>component " message_string"</i></p> <p>component is the component in the router which generated the message. Possible components are:</p> <p>GBF
ROOT
PPM2
SNMP
LC
LDAP
AUtoStartUp
INIT
DHCP
SCHEL</p> <p>message string is the message being logged. Examples of the message strings:</p> <p>" ISDN SwitchType changed; line is being initialized..."</p> <p>" DO flush_dmac"</p> <p>" Wed Feb 3 04:43:11 1999 Circuit UP
LMAC 0A804440860A LSAP 04 RMAC
400040000002 RSAP 04 IP
10.5.146.250 "</p> | LogINfo | AuditLog service control & config |
| WEBLink | 1400 | WEBLink login success | " Web Link Login Successful" | LogNOtice | AuditLog service control & config |

| Service | Msg ID | Message name/type | STRING being logged | Severity | CONTROL Parameters |
|---------|--------|-----------------------------|--|------------|-----------------------------------|
| WEblink | 1401 | WEblink session termination | " Web Link Session Terminated" | LogNOtice | AuditLog service control & config |
| WEblink | 1402 | WEblink login failure | " Login Failed" | LogALert | AuditLog service control & config |
| WEblink | 1403 | WEblink MIB command | SET request for <i>snmp_oid</i> Value = <i>value</i>
Example:
SET request for 1.3.6.1.2.1.1.6.0 Value = my_cube | LogWArning | AuditLog service control & config |
| WEblink | 1404 | HTTP server login success | HTTP Login Successful | LogALert | AuditLog service control & config |
| System | 9999 | Console command | any configuration change command that was executed successfully
Example:
UI cmd "setd !4 -PATH CONTROL = e"
"me -PATH" executed | LogWArning | AuditLog service control & config |



REGULAR EXPRESSIONS

This appendix describes the regular expressions used for creating and displaying AS-path-based filters (AsPolicyAll, AsPolicyExt, AsPolicyInt, AsPolicyPeer, and DisplayFilter parameters) in the BGP Service and for altering SHow and SHowDefault displays with the GREP command.

You can use regular expressions to specify a general string. This general string can then be used for pattern matching.

A regular expression is a formula for generating a set of strings. If a particular string can be generated by a given regular expression, that string and regular expression match. In many ways, a regular expression is a program, and the regular expression matches the strings the program generates.

A regular expression consists of different components described in Table 130, each of which is used to build the regular expression string-generating program.

Table 130 Regular Expression Components

| Regular Expression | Function |
|--------------------|--|
| c | Use an ordinary ASCII character (excluding the special characters) to match that character. For example, c matches a lowercase c. |
| . | Use a period (.) to match any character except NEWLINE. |
| * | Use a regular expression followed by an asterisk (*) to match zero or more occurrences of the one-character regular expression. If there is any choice, the longest leftmost string that permits a match is chosen. |
| + | Use a regular expression followed by plus sign (+) to match one or more occurrences of the one-character regular expression. If there is any choice, the longest leftmost string that permits a match is chosen. |
| ? | Use a regular expression followed by a question mark (?) to match zero or one occurrences of the one-character regular expression. If there is any choice, the longest leftmost string that permits a match is chosen. |
| | Use two regular expressions separated by vertical bar () or NEWLINE to match either the first or the second (logical OR operation). |
| () | Use a regular expression enclosed in parentheses to match the regular expression. |
| \. | Use a backslash (\) followed by any special character (period, asterisk, left square bracket, backslash) to match the special character. |
| * | Use a backslash (\) followed by any special character (period, asterisk, left square bracket, backslash) to match the special character. |
| [| These characters are special except when they are enclosed within square brackets ([]). For example, * matches an asterisk (*). |
|] | These characters are special except when they are enclosed within square brackets ([]). For example, * matches an asterisk (*). |

(continued)

Table 130 Regular Expression Components (continued)

| Regular Expression | Function |
|---|--|
| [string] | <p>Use a non-empty string of characters enclosed in square brackets to match any one character in that string.</p> <p>If the first character of the string is a caret (^), the one-character regular expression matches any character except NEWLINE and the remaining characters in the string. The caret has this special meaning only if it occurs first in the string.</p> <p>The hyphen (-) indicates a range of consecutive ASCII characters; for example, [0-9] is equivalent to [0123456789]. The hyphen loses this special meaning if it occurs first (after an initial ^, if any) or last in the string.</p> <p>The right square bracket (]) does not terminate a string when it is the first character within it (after an initial ^, if any); for example, []a-f] matches either a right square bracket or one of the letters a through f inclusive.</p> <p>The period, asterisk, left square bracket, and backslash represent themselves within such a string of characters.</p> |
| <p>Concatenation: The remaining regular expressions are for concatenation, a regular expression that matches the concatenation of the strings matched by each component of the regular expression.</p> | |
| \< | Use the sequence \< in a regular expression to constrain the one-character regular expression immediately following it only to match something at the beginning of a "word;" for example, either at the beginning of a line, or just before a letter, digit, or underline and after a character not one of these. |
| \> | Use the sequence \> in a regular expression to constrain the one-character regular expression immediately following it only to match something at the end of a "word;" for example, either at the end of a line, or just before a character, which is neither a letter, digit, nor underline. |
| \(and \) | Use a regular expression enclosed between the character sequences \(and\) to match whatever the unadorned regular expression matches. |
| \{m\} | Use a regular expression followed by \{m\}, \{m,\}, or \{m,n\} to match a range of occurrences of the regular expression. |
| \{m,\} | The values of m and n must be non-negative integers less than 256. |
| \{m,n\} | <p>\{m\} matches exactly m occurrences.</p> <p>\{m,\} matches at least m occurrences.</p> <p>\{m,n\} matches any number of occurrences between m and n inclusive.</p> <p>Whenever a choice exists, the regular expression matches as many occurrences as possible.</p> |
| \n | <p>Use the expression \n to match the same string of characters that was matched by an expression enclosed between \(and\) earlier in the same regular expression.</p> <p>n is a digit; the subexpression specified begins with the nth occurrence of \ (counting from the left).</p> <p>For example, the expression ^\(.*)\1\$ matches a line consisting of two repeated appearances of the same string.</p> |
| ^ | Use the caret (^) at the beginning of a regular expression to constrain the regular expression to match an initial segment of a line. |
| \$ | <p>Use the dollar sign (\$) at the end of a regular expression to constrain the regular expression to match a final segment of a line.</p> <p>For example, ^entire regular expression \$ constrains the regular expression to match the entire line.</p> |

AS Filter Examples

This section provides examples of autonomous system (AS) filters using regular expressions. The following syntax is used.

```
ADD -BGP AsFilter <AsfilterID> "<regular expression>"
```



Blank spaces are represented here as underscores (_). When two spaces are shown together, a space has been inserted between the underscores, for example _ _ . You must enter a blank space for each underscore shown in the examples.

Example 1 To create filter 1 that identifies an AS-path attribute containing AS25, enter:

```
ADD -BGP ASFilter 1 "_25_"
```

Example 2 To create filter 2 that identifies an AS-path attribute containing AS35 and AS50 (in this order), enter:

```
ADD -BGP ASFilter 2 "_35_.*_50_"
```

The ".*" indicates a single character followed by any number of unspecified characters.

Example 3 To create filter 3 that identifies an AS-path attribute containing AS35 and AS50 (in any order), enter:

```
ADD -BGP ASFilter 3 "_35_.*_50_|_50_.*_35_"
```

The "|" indicates a logical OR operation.

Example 4 To create filter 4 that identifies an AS-path attribute containing the AS sequence <AS5, AS46, AS32>, enter:

```
ADD -BGP ASFilter 4 "<_5_ _46_ _32_>"
```

Example 5 To create filter 5 that identifies an AS-path attribute containing the AS set [AS5, AS46, AS32], enter:

```
ADD -BGP ASFilter 5 "[_5_ _32_ _46_]"
```

AS sets are always sorted from lowest AS to highest AS.

GREP Command Examples

This section provides examples of commands used with the GREP filter and regular expressions. The following syntax is used:

```
<COMMAND> <parameters> [<options>] | GREP [-v] [-i] <grep pattern>
```

You can use the GREP command with the SHow and SHowDefault commands to alter the output to display the information you specify in the GREP pattern.



You cannot apply multiple GREP commands to a single UI command. The following command to show all IP routes that have the number 192 and 128 is not supported:

```
SHow -IP AllRoutes | GREP 192 | GREP 128
```

Example 1 To display all IP routes, you normally enter the SHow command. To show only those routes that have the number 152 in them, you can pipe the SHow or SHowDefault output to the GREP command by entering:

```
SHow -IP AllRoutes | GREP 152
```

The following display appears:

```
129.213.152.0    255.255.252.0    129.213.200.109    1    UP    RIP
```

The number 152 in regular expression form is a string of ASCII characters that are matched, generating the information you specified.

Example 2 To display all IP routes that do not have the number 152 in them, pipe the output to the GREP command by entering:

```
SHow -IP AllRoutes | GREP -v 152
```

The following display appears:

```
----- IP Routing Table -----
Total Routes = 12, Total Direct Networks = 1
Destination      Mask              Gateway           Metric   Status   TTL   Source
0.0.0.0          0.0.0.0          129.213.200.109  2        UP       170   RIP
                 129.213.200.103  3        UP       150   RIP
129.213.16.0     255.255.252.0    129.213.200.109  1        UP       170   RIP
                 129.213.200.108  1        UP       170   RIP
129.213.32.0     255.255.252.0    129.213.200.109  1        UP       170   RIP
129.213.48.0     255.255.252.0    129.213.200.109  1        UP       170   RIP
                 129.213.200.102  1        UP       170   RIP
129.213.72.0     255.255.252.0    129.213.200.109  1        UP                RIP
129.213.96.0     255.255.252.0    129.213.200.103  1        UP                RIP
129.213.200.0    255.255.252.0    129.213.203.16   0        UP       --    Connected
129.213.240.0    255.255.252.0    129.213.200.109  1        UP                RIP
```

The -v option for GREP performs the NOT (or invert) operation on the display information and lists all the IP routes that do not have the number 152 in them. For descriptions of the GREP options, refer to the GREP command description in the Commands chapter in *Reference for Enterprise OS Software*.

Example 3 To display all IP routes containing the number 72 or 96 in them, pipe the output to the GREP command by entering:

```
How -IP AllRoutes | GREP 72 | 96
```

The following display appears:

```
129.213.152.0    255.255.252.0    129.213.200.109    1    UP    RIP
```

The first "|" represents the pipe to the GREP command. The "|" between the numbers is the regular expression for logical OR. In this example, the output matches any string that contains the numbers 72 or 96.

X.3 PARAMETERS AND PAD PROFILES

This appendix provides the X.3-to-TERM Service session parameter mappings as well as the 3Com implementation of the Consultive Committee for International Telegraph and Telephone (CCITT) Simple Standard packet assembler/disassembler (PAD) Profile Number 90 parameter settings.

X.3-to-TERM Service Parameter Equivalence

Table 131 lists the standard X.3 profile parameters and equivalent parameters, that currently operate on a bridge/router which functions as a connection service gateway. For information on X.3 profile parameters, refer to CCITT Recommendations X.3 and X.29.

Table 131 X.3-to-TERM Service Parameter Equivalence

| PAD Parameter | X.3 Profile Parameters | TERM Service Parameters | Default Setting |
|---------------|--|-------------------------|---|
| Parameter 1 | PAD recall using a character | ECMChar | ^A |
| Parameter 2 | Echo | ECHOData | ON |
| Parameter 3 | Selection of data forwarding character | DataForward | CR, ESC, EDiting, Term, FormEf, COntrOl |
| Parameter 4 | Selection of idle timer delay | IdleTimer | 1 |
| Parameter 5 | Ancillary device control | FlowCtrlFrom | Xon_Xoff |
| Parameter 6 | Control of PAD service signal | None | Not applicable (cannot be configured by user) |
| Parameter 7 | PAD on receipt of break | BReakAction | InBand |
| Parameter 8 | Discard output | FlushVC | OFF |
| Parameter 9 | Padding after carriage return | None | None |
| Parameter 10 | Line folding | None | Not applicable (cannot be configured by user) |
| Parameter 11 | Binary speed of start-stop DTE | BAud | 9600 |
| Parameter 12 | Flow control of the PAD | FlowCtrlTo | Xon_Xoff |
| Parameter 13 | Linefeed insertion after carriage return | LFInsertion | None |
| Parameter 14 | Padding after linefeed | None | None |
| Parameter 15 | Editing | LocalEDit | OFF |
| Parameter 16 | Character delete | ERase | ^? |
| Parameter 17 | Line delete | LineERase | ^X |
| Parameter 18 | Line display | ReprintLine | ^R |
| Parameter 19 | Editing PAD service signals | None | Not applicable (cannot be configured by user) |
| Parameter 20 | Echo mask | ECHOMask | None |
| Parameter 21 | Parity treatment | PARItY | None |
| Parameter 22 | Page wait | None | Not applicable (cannot be configured by user) |

CCITT Simple Standard PAD Profile

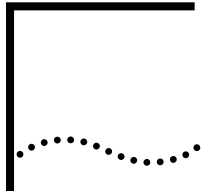
Table 132 lists the default values of the CCITT Simple Standard PAD Profile. You can select these profiles to use with incoming and outgoing extended connections. With outgoing extended connections, you can alter the settings of these parameters to create customized profiles (configuration files) as described in the Configuring Connections for Outgoing Calls chapter.



The 3Com implementation of the CCITT Simple Standard PAD Profile is based on the CCITT Simple Standard PAD Profile Number 90, but does not exactly match the official CCITT definition. In the 3Com implementation of Profile Number 90, the value of PAD parameter number 19 has been changed to 2 and parameter number 6 has been changed to 5.

Table 132 CCITT Simple Standard PAD Profile of CCITT PAD Profile 90

| PAD Parameter Number | PAD Parameter Name | Value | Meaning |
|----------------------|--|-------|--|
| 1 | PAD recall using a character | 1 | Escape from data transfer |
| 2 | Echo | 1 | Local echo |
| 3 | Selection of data forwarding character | 126 | All characters in column 0 and 1 and character DEL |
| 4 | Selection of idle timer delay | 0 | No idle timer delay |
| 5 | Ancillary device control | 1 | Use of XON/XOFF (data transfer) |
| 6 | Control of PAD service signal | 5 | Pad service signals and the prompt PAD service signal are transmitted in the standard format |
| 7 | PAD on receipt of break | 2 | Reset |
| 8 | Discard output | 0 | Normal data delivery |
| 9 | Padding after carriage return | 0 | No padding after carriage return |
| 10 | Line folding | 0 | No line folding |
| 11 | Binary speed of start-stop mode DTE | 14 | Baud rate (9600) |
| 12 | Flow control of the PAD | 1 | Use of XON/XOFF for flow control |
| 13 | Linefeed insertion after carriage return | 0 | No linefeed insertion |
| 14 | Padding after linefeed | 0 | No padding after linefeed |
| 15 | Editing | 0 | No editing in the data transfer state |
| 16 | Character delete | 127 | Character DEL |
| 17 | Line delete | 24 | ASCII 18 |
| 18 | Line display | 18 | ASCII 12 |
| 19 | Editing PAD service signals | 2 | Editing PAD service signals for display terminals |
| 20 | Echo mask | 0 | All characters echoed |
| 21 | Parity treatment | 0 | None |
| 22 | Page wait | 0 | Disabled |



WIDE AREA NETWORK SETUP INFORMATION

This appendix provides information to help you set up your wide area serial ports.

NETBuilder II I/O Module Placement

Do not insert a token ring I/O module into the NETBuilder II chassis directly above a HSS V.35 3-Port module with part number 06-0107-000.



CAUTION: *This module placement can cause overheating. Any other placement of the token ring I/O module and the HSS V.35 3-Port module is acceptable.*

This module placement problem does not occur with HSS V.35 3-Port module part number 06-0124-xxx.

T3 Plus Interoperability

In the following manuals for the BMX45S Bandwidth Manager from T3 Plus Network, Inc., the installation instructions incorrectly describe how to configure the BMX45S to work with a 3Com NETBuilder bridge/router:

- *BMX45 T3 Bandwidth Manager: User Manual (DOS/Windows)*, part number 010-10148-0001, Rev. E
- *BMX45 T3 Bandwidth Manager: User Manual (UNIX/SNMP NMS)*, part number 010-10373-0001, Rev. B

In these manuals, Table 2-3, "Typical Strapping Option Requirements," describes how to configure BMX45S with 3Com, NCS, Cisco, and Bay Networks products. The 3Com NETBuilder description is incorrect; the transmit clock setting should be set to EXTERNAL.

HSS Port Utilization Percentage

The percentage of utilization displayed for HSS, HSSI, and HSS V.35 3-Port WAN ports is based on a full-duplex link. For example, a 64-kbps circuit can transmit and receive 64 kbps simultaneously. If this link were transmitting at 64 kbps and receiving nothing, the percentage of utilization would be 50 percent.

To display utilization information, enter:

```
SHow -SYS STATistics -PATH
```

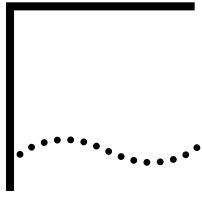
Serial Line Connectivity

The following sections provide information about serial line connections, clocking, cables, and data rates.

External Device Connections

When configuring your NETBuilder II bridge/router for remote communications with external clocking devices, 3Com recommends that you set the -PATH CONNector and CLock parameters to match the external devices before physically connecting the two devices.

- External Device Cable Length** At serial line data rates greater than 56 kbps on RS-232 interfaces, 3Com recommends using cables less than 25 feet long. The HSS V.35 3-Port WAN interface includes an 8-foot external adapter cable. A cable attached to this adapter should be no more than 5 feet long.
- Serial Line Clocking** When using a NETBuilder II bridge/router or SuperStack II NETBuilder bridge/router with a modem or channel service unit/data service unit (CSU/DSU), you must determine which piece of equipment provides the clocking signal. For a bridge/router using an external device over dial-up lines, the external device must provide clocking. For a bridge/router using an external device over a leased line, either device can provide clocking.
- If you configure an external device to provide clocking to a bridge/router, the external device must use the return clock provided by the bridge/router. Neither the external device nor the cable used to connect it to the bridge/router should loop back the transmit clock to the receive clock.
- If you connect two SuperStack II NETBuilder bridge/routers, or a SuperStack II NETBuilder bridge/router to a NETBuilder II bridge/router with an HSS 3-Port WAN interface, you must use a modem eliminator and set the -PATH CLock parameter to External on both devices. Contact your 3Com supplier for a suggested list of modem eliminators.
- Synchronizing the Network Clock** The Network Time Protocol (NTP) provides a standard mechanism to synchronize the computer clock in the distributed network. NTP support in NETBuilder II bridge/routers is based on RFC1305. The NTP service is a client/server model. NTP can be applied to clients as well as servers. As a client, NTP periodically polls the designated time server for time of day and processes the reply to determine the local clock update. As a server, NTP responds with its local time to the NTP client. The client retrieves the network clock from the most accurate time server and the server provides the time source to its private network. For information about setting NTP parameters, see the NAT Service Parameters chapter in *Reference for Enterprise OS Software*.
- Serial Line Supported Data Rates** NETBuilder software supports serial line data rates above 2,048 kbps for external clocking devices only. If the selected data rate is above 2,048 kbps with internal clocking (Test Mode) specified, the actual data rate used by the bridge/router is 2,048 kbps.
- When using external clocking devices, set the data rate of the serial line on the bridge/router as close as possible to the external device. Packet processing is optimized to this value. Statistical reporting of line utilization is based on the data rate you configure.



APPN CONFIGURATION EXAMPLES

This appendix provides examples of how to configure Advanced Peer-to-Peer Networking (APPN) on the 3Com network node so that sessions to and from other commonly used IBM platforms can take place. For information on basic APPN configuration steps, see the Configuring APPN Intermediate Session Routing chapter.

Unless otherwise noted, the examples in this appendix assume that the NETBuilder II bridge/router and the corresponding IBM platforms are configured for Intermediate Session Routing (ISR) only.

AS/400 Configuration

This section provides examples of how to configure the 3Com APPN network node to communicate to and from an AS/400 in both token ring and Frame Relay environments.



If you change transmission group (TG) characteristics using the LinkStaCHar parameter on the NETBuilder II bridge/router, you must also change the corresponding TG characteristics on the AS/400 to match. If you change TG characteristics on the AS/400 for a link to the NETBuilder II system, you must define a link station on the NETBuilder II system to the AS/400 and modify the TG class to match.

Example 1: Token Ring Over Physical Ports

Figure 462 is an example of an AS/400 and a 3Com NETBuilder II system connected in a token ring environment. Table 133 lists the parameters that must be configured on each platform if the AS/400 initiates the connection to the NETBuilder II system. Table 134 lists the parameters that must be configured differently on each platform if the NETBuilder II system initiates the connection to the AS/400 (all other parameters are configured as shown in Table 134).

If the AS/400 initiates the connection, the AS/400 initial connection setting must be set to DIAL, and the adjacent link station to the AS/400 does not need to be configured on the NETBuilder II system. If the NETBuilder II system initiates the session request, then the adjacent link station must be configured, and the initial connection setting on the AS/400 must be set to ANSWER.

When configuring the remote media access control (MAC) addresses, you configure MAC addresses in noncanonical format on the AS/400 and canonical

format on the NETBuilder II system. In the tables, the MAC addresses are shown in the respective formats that must be used on each platform.

Figure 462 Token Ring Configuration Between NETBuilder II System and AS/400

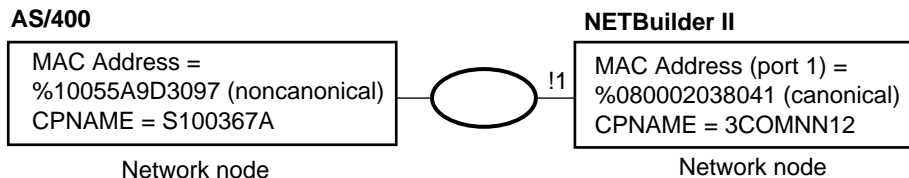


Table 133 AS/400 Parameters to Initiate Token Ring Connection with NETBuilder II System (ISR Only)

| AS/400 Parameters for Connection with 3Com Network Node | NETBuilder II Commands for Responding to Connection Requests from AS/400 Network Node |
|---|---|
| <p>Change Line Description:
 Line Description: TOKENRING1*
 Local adapter address: 10005A9D3097</p> <p>Change Controller Description:
 Controller Description: 3COMNN12
 Option: *Basic
 Category of Controller: *APPC
 Link type: *LAN
 Maximum frame size: 1033‡
 APPN/HPR capable: *NO
 Active switched line: TOKENRING1*
 Remote network identifier: US3COMHQ
 Remote control point: 3COMNN12
 Exchange Identifier: E06xxxx†
 Initial connection: *DIAL
 Dial initiation: *LINKTYPE
 LAN remote adapter address: 100040C00182</p> | <p>Enable the port:
 SETDefault !1 -PORT CONTROL = Enabled</p> <p>Set APPN parameters:
 SETDefault -APPN LocalNodeName = US3COMHQ.3COMNN12 xxxxx†
 SETDefault !1 -APPN PortDef = LLC2 1033‡ 0 80 HPR=No</p> |

* The line description is the name you assign to the line. The line description must match the name used for the active switched line parameter.

† 3Com network nodes send the exchange identifier E06xxxx. The digits xxxx are set in LocalNodeName, and the AS/400 exchange identifier must match. If you do not configure the ID number, it defaults to 00000.

‡ The maximum frame size setting on the AS/400 should match the maximum BTU size value on the NETBuilder II network node. However, if these values do not match, the value is negotiated and the lower value is used.

Table 134 NETBuilder II Parameters to Initiate Token Ring Connection with AS/400 (ISR Only)

| AS/400 Parameters for Responding to Connection Requests from 3Com Network Node | NETBuilder II Commands for Connection with AS/400 Network Node |
|--|--|
| <p>Change Controller Description:
 Initial connection: *ANS</p> | <p>Define AS/400 as an adjacent link station through token ring:
 ADD !1 -APPN AdjLinkSta NN 1033 NC10005A9D3097* 04† HPR=NO</p> |

* The MAC address here matches the Local Adapter Address of the AS/400 in Table 133. The address is entered here in noncanonical format.

† SAP 04 represents the SAP of the AS/400. Many IBM devices use SAP 04. To verify the correct SAP, consult IBM documentation.

If the AS/400 is an end node, then a different configuration is required. Table 135 lists the different configuration necessary on both sides if the AS/400 is an end node. If the AS/400 is an end node, for example, then you do not have to configure the AS/400 as an adjacent link station on the 3Com network node because the AS/400 calls into the 3Com network node. Unless listed here, the

configuration on the AS/400 is the same as in Table 134, since the AS/400 initiates the connection with the network node.

Table 135 Settings if AS/400 is an End Node

| AS/400 Acting as End Node | NETBuilder II Commands |
|--|------------------------|
| Change Network Attributes:
Node Type = *ENDNODE
Network node servers
Server network ID: US3COMHQ
Control point name: 3COMNN12 | None |

The previous examples assume that both the NETBuilder II bridge/router and the AS/400 are both ISR nodes only. Table 136 lists how you would enter the commands differently to configure High Performance Routing (HPR) support for both nodes.

Table 136 Settings to Configure Differently for Both Nodes to Support HPR

| Parameters for AS/400 | NETBuilder II Commands |
|---|--|
| Change Controller Description:
APPN/HPR capable: *YES | SETDefault !1 -APPN PortDef = LLC2 1033 0 80 HPR=Yes
ADD !1 -APPN AdjLinkSta NN 1033 NC10005A9D3097 HPR=Yes |

Example 2: Frame Relay over Physical Ports

Figure 463 is an example of an AS/400 and a 3Com NETBuilder II system connected in a Frame Relay environment using physical ports. Table 137 lists the parameters that must be configured on each platform if the AS/400 initiates the connection to the NETBuilder II system. Table 138 lists the parameters that must be configured differently on each platform if the NETBuilder II system initiates the connection to the AS/400 (all other parameters would be configured the same as shown in Table 137).

Figure 463 Frame Relay Configuration Between NETBuilder II System and AS/400 (Physical Port)

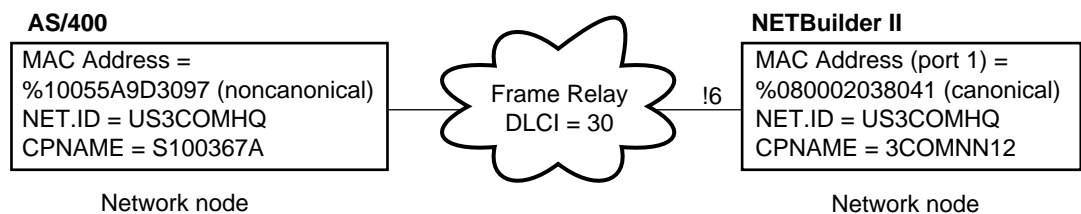


Table 137 AS/400 Parameters to Initiate Frame Relay Connection with NETBuilder II (Physical Ports)

| AS/400 Parameters | NETBuilder II Commands |
|--|---|
| Network Interface (NWI) Description:
Network interface description: FRAMERELAY
Category of NWI: *FR
Option: *BASIC
Line speed: 1536000
LMI mode: *NONE | Set Path parameters:
SETDefault !6 -PATH BAud = 1536
SETDefault !6 -PATH CONTrol = Enabled
Set up port for Frame Relay:
SETDefault !6 -PORT OWNer = FrameRelay
SETDefault !6 -FR CONTrol = NoLMI |

Table 137 AS/400 Parameters to Initiate Frame Relay Connection with NETBuilder II (Physical Ports) (continued)

| | |
|---|---|
| <p>Change Line Description:
 Line Description: FR1
 Attached nonswitched NWI: FRAMERELAY
 DLC Identifier: 30
 Exchange Identifier: 0560367A*
 Maximum frame size: 1033</p> <p>Change Controller Description:
 Controller Description: 3COMNN12DIAL
 Option: *Basic
 Category of Controller: *APPC
 Link type: *FR</p> <p>Change Controller Description: (cont.)
 Maximum frame size: 1033
 Remote network identifier: US3COMHQ
 Exchange Identifier: E0600000†‡
 Initial connection: *DIAL
 Dial initiation: *LINKTYPE</p> | <p>Set data link type to be Frame Relay:
 SETDefault !6 -APPN PortDef = FR 1033 0 80</p> <p>Enable the port:
 SETDefault !6 -PORT CONTROL = Enabled</p> <p>Set APPN parameters:
 SETDefault -APPN LocalNodeName = US3COMHQ.3COMNN12 00000†</p> |
|---|---|

* This is the Exchange Identifier for the AS/400.
 † The last five hex digits in these two entries must match.
 ‡ This is the exchange identifier for the NETBuilder II system.

Table 138 NETBuilder II Parameters to Initiate Frame Relay Connection with AS/400 (Physical Ports)

| AS/400 Parameters | NETBuilder II Commands |
|--|--|
| <p>Change Controller Description:
 Initial connection: *ANS</p> | <p>Define the AS/400 as an adjacent link station through Frame Relay:
 ADD !6 -APPN AdjLinkSta NN 1033 30 4 USCOMHQ.S100367A 0367A*</p> |

* The last five hex digits must match the Exchange Identifier in the AS/400 line description.

Table 139 lists parameters to set differently from those shown in Table 137 if you are using a modem, switch, or modem eliminator. For this example, set the baud rate in the modem eliminator to 64000 (the actual speed depends on the modem, switch, or modem eliminator being used).

Table 139 Parameters for Modem, Modem Eliminator, or Switch that Provides Clocking

| AS/400 Parameters | NETBuilder II Commands |
|---|---|
| <p>Change NWI Description:
 Line speed: 64000*</p> | <p>Set baud rate to match that of modem, modem eliminator, or switch:
 SETDefault !6 -PATH BAud = 64*</p> <p>Use clock from modem eliminator:
 SETDefault !6 -PATH CLock = External</p> |

* The line speed setting on the AS/400 should match the path baud rate setting on the NETBuilder II network node.

Example 3: Frame Relay over Virtual Ports

Figure 464 is an example of an AS/400 and a 3Com NETBuilder II system connected in a Frame Relay environment using physical ports. Table 140 lists the parameters that must be configured on each platform if the AS/400 initiates the connection to the NETBuilder II system. Table 141 lists the parameters that must

be configured differently if the NETBuilder II system initiates the connection to the AS/400.

Figure 464 Frame Relay Configuration Between NETBuilder II System and AS/400 (Virtual Port)

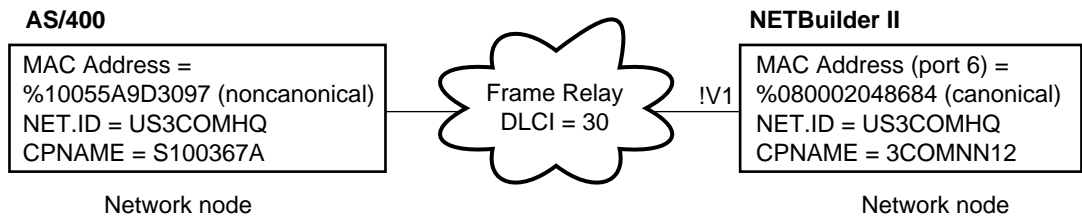


Table 140 AS/400 Parameters to Initiate Frame Relay Connection with NETBuilder II (Virtual Ports)

| AS/400 Parameters | NETBuilder II Commands |
|--|--|
| <p>Network Interface (NWI) Description:
 Network interface description: FRAMERELAY
 Category of NWI: *FR
 Option: *BASIC
 Line speed: 1536000
 LMI mode: *NONE</p> <p>Change Line Description:
 Line Description: FR1
 Attached nonswitched NWI: FRAMERELAY
 DLC Identifier: 30
 Exchange Identifier: 0560367A*
 Maximum frame size: 1033</p> <p>Change Controller Description:
 Controller Description: 3COMNN12DIAL
 Option: *Basic
 Category of Controller: *APPC
 Link type: *FR
 Maximum frame size: 1033
 Remote network identifier: US3COMHQ</p> | <p>Set Path parameters:
 <code>SETDefault !6 -PATH BAud = 1536</code>
 <code>SETDefault !6 -PATH CLock = TestMode</code></p> <p>Set up port for Frame Relay:
 <code>SETDefault !6 -PORT OWner = FrameRelay</code>
 <code>SETDefault !6 -FR CONTROL = NoLMI</code></p> <p>Set virtual port:
 <code>ADD !V1 -PORT VirtualPort 6@30</code></p> <p>Set port type to be Frame Relay:
 <code>SETDefault !6 -APPN PortDef = FR 1033 0 80</code></p> <p>Enable the port:
 <code>SETDefault !V1 -PORT CONTROL = Enabled</code></p> <p>Set APPN parameters:
 <code>SETDefault -APPN LocalNodeName = US3COMHQ.3COMNN12 0000†</code></p> |

* This is the Exchange Identifier for the AS/400.

† The last five hex digits in these two entries must match.

Table 141 NETBuilder II Parameters to Initiate Frame Relay Connection with AS/400 (Virtual Ports)

| AS/400 Parameters | NETBuilder II Commands |
|--|--|
| <p>Change Controller Description:
 Initial connection: *ANS</p> | <p>Define the AS/400 as an adjacent link station through Frame Relay:
 <code>ADD !V1 -APPN AdjLinkSta NN 1033 30 4 USCOMHQ.S100367A 0367A*</code></p> |

* The last five hex digits must match the Exchange Identifier in the AS/400 line description.

IBM PC Support/400 Example

This section provides examples of how to configure the 3Com APPN network node to communicate to and from PCs running PC Support/400.

Example 4: Setting Up Connections with a DOS PC

Figure 465 is an example in which a DOS PC client is trying to access a logical unit on an AS/400 server, with the NETBuilder II system acting as the network node server for the PC. Table 142 lists the commands you need to configure on the PC and on the NETBuilder II system for the PC to initiate the BINDs.

Figure 465 Token Ring Configuration Between NETBuilder II System and DOS PC

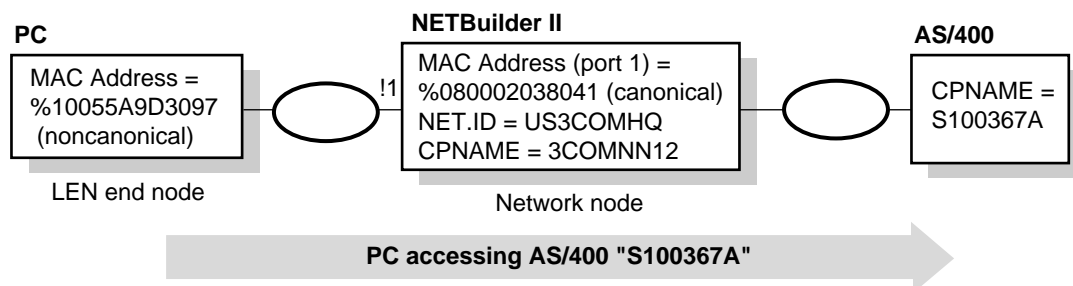


Table 142 DOS PC Configuration to Initiate Connection with NETBuilder II System

| DOS PC Commands for initiating Connection with Server via 3Com Network Node | NETBuilder II Commands for Responding to Connection Requests from DOS PC End Node |
|--|--|
| SFLR <u>1,I,S100367A</u> *
UPDT I:\QIWSFL2,C:\PCS,S,,,PC Support/400
RTYP ITRN
RTLN US3COMHQ.USER†
TRLI <u>S100367A</u> , 100040C00182‡
ADRS PUBS**, S100367A†† | Set data link type for token ring:
<code>SETDefault !1 -APPN PortDef = LLC2 1033 0 80</code>
Enable the port:
<code>SETDefault !1 -PORT CONTROL = Enabled</code>
Set APPN parameters:
<code>SETDefault -APPN LocalNodeName = US3COMHQ.3COMNN12 0000</code> |

* The value underlined on line SFLR must match the value underlined on line TRLI.

† US3COMHQ.USER is the LU name for PC Support/400.

‡ This is the MAC address of the NETBuilder II system in noncanonical format.

**This is the name of a second AS/400.

††This line is used only if the NETBuilder II system is connecting to more than one AS/400.

Configuration for DLUs/DLUr

This section provides an example of the Virtual Telecommunications Access Method (VTAM) host configuration parameters and how they must match parameters on the Dependent Logical Unit Requester (DLUr) and physical unit (PU) 2x nodes.



This example is for Intermediate Session Routing only. For information on configuring HPR for VTAM, see the IBM document VTAM V4.3: High Performance Routing (HPR) Early User Experiences (SG24-4507-00).

Figure 466 is an example in which a VTAM host is serving as the dependent LU server (DLUs) for a PU 2.x node with dependent LUs. The NETBuilder II bridge/router serves as the network node DLUr.

Figure 466 VTAM Host Configuration for DLUs/DLUr

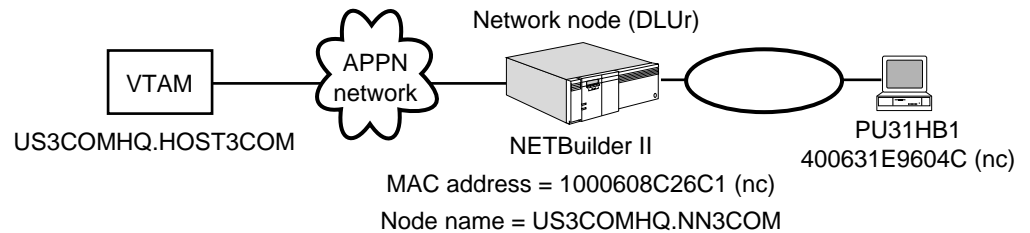


Table 143 is an example configuration with a VTAM host as a DLUs node and a network node acting as the DLUr. The table depicts how parameters must match to make the configuration work. This configuration assumes that VTAM is configured for APPN and is at least level 4.2 or higher and that DLUr and DLUs can establish LU6.2 sessions with each other. Both DLCADDR statements are required for DLUs initiated activation.

Table 143 VTAM Configuration for DLUr

| PU Definition on VTAM | DLUr Link Station on DLUr Node | Host Link Definition on PU 2.x |
|---|--|--|
| PU31HB1 PU ADDR=01
ANS=CONT,
DLOGMOD=D4C32782,
...
IDBLK=05D,
IDNUM=24001,
...
HB1PATH PATH PID=1,
DLURNAME=PEBBLE,
DLCADDR=(1,C,INTPU),
DLCADDR=(2,X,05D24001) | ADD !1 D lurLinkStation 1033 Ncmac 400031E9604C
PU31HB1 Dlus = HOST3COM | Local node ID = 05D24001
LAN Destination Address = 1000608C26C1 (nc)
MAC Address of DLUr |

APPN Sense Codes

This section lists APPN sense codes. Table 144 lists the APPN primary return sense codes.

Table 144 APPN Primary Return Sense Codes

| Sense Codes | Hex |
|--------------------|----------|
| OK | (0x0000) |
| PARAMETER_CHECK | (0x0100) |
| STATE_CHECK | (0x0200) |
| ALLOCATION_ERROR | (0x0300) |
| DEALLOC_ABEND | (0x0500) |
| DEALLOC_ABEND_PROG | (0x0600) |

Table 144 APPN Primary Return Sense Codes

| Sense Codes | Hex |
|---------------------------|----------|
| DEALLOC_ABEND_SVC | (0x0700) |
| DEALLOC_ABEND_TIMER | (0x0800) |
| DEALLOC_NORMAL | (0x0900) |
| PROG_ERROR_NO_TRUNC | (0x0C00) |
| PROG_ERROR_TRUNC | (0x0D00) |
| PROG_ERROR_PURGING | (0x0E00) |
| CONV_FAILURE_RETRY | (0x0F00) |
| CONV_FAILURE_NO_RETRY | (0x1000) |
| SVC_ERROR_NO_TRUNC | (0x1100) |
| SVC_ERROR_TRUNC | (0x1200) |
| SVC_ERROR_PURGING | (0x1300) |
| UNSUCCESSFUL | (0x1400) |
| CNOS_PARTNER_LU_REJECT | (0x1800) |
| CONVERSATION_TYPE_MIXED | (0x1900) |
| NODE_STOPPING | (0x1A00) |
| NODE_NOT_STARTED | (0x1B00) |
| CANCELLED | (0x2100) |
| BACKED_OUT | (0x2200) |
| CONVERSATION_ENDED | (0x4200) |
| THREAD_BLOCKING | (0xF006) |
| INDICATION | (0x0210) |
| ACTIVATION_FAIL_RETRY | (0x0310) |
| ACTIVATION_FAIL_NO_RETRY | (0x0410) |
| LU_SESS_LIMIT_EXCEEDED | (0x0510) |
| FUNCTION_NOT_SUPPORTED | (0x0610) |
| TP_BUSY | (0x02F0) |
| COMM_SUBSYSTEM_ABENDED | (0x03F0) |
| COMM_SUBSYSTEM_NOT_LOADED | (0x04f0) |
| UNEXPECTED_SYSTEM_ERROR | (0x11F0) |
| INVALID_VERB | (0xFFFF) |

Table 145 lists the APPN secondary return sense codes.

Table 145 APPN Secondary Return Sense Codes

| Sense Codes | Hex |
|-----------------------------|---------------|
| ALLOCATE_NOT_PENDING | (0x09050000L) |
| ALLOCATION_FAILURE_NO_RETRY | (0x04000000L) |
| ALLOCATION_FAILURE_RETRY | (0x05000000L) |
| INVALID_NODE_TYPE_FOR_HPR | (0xC8020000L) |
| BAD_CONV_ID | (0x02000000L) |
| BAD_CONV_TYPE | (0x11000000L) |
| BAD_ERROR_DIRECTION | (0x05010000L) |

Table 145 APPN Secondary Return Sense Codes (continued)

| Sense Codes | Hex |
|-----------------------------|---------------|
| BAD_LL | (0xF1000000L) |
| BAD_REMOTE_LU_ALIAS | (0x03000002L) |
| BAD_RETURN_CONTROL | (0x14000000L) |
| BAD_RETURN_STATUS_WITH_DATA | (0xD7000000L) |
| BAD_SECURITY | (0x13000000L) |
| BAD_SYNC_LEVEL | (0x12000000L) |
| BAD_TP_ID | (0x01000000L) |
| BAD_TYPE | (0x50020000L) |
| BO_NO_RESYNC | (0x00002408L) |
| BO_RESYNC | (0x01002408L) |
| CONFIRMED_BAD_STATE | (0x41000000L) |
| CONFIRM_BAD_STATE | (0x32000000L) |
| CONFIRM_NOT_LL_BDY | (0x33000000L) |
| CONFIRM_ON_SYNC_LEVEL_NONE | (0x31000000L) |
| COS_NAME_NOT_DEFD | (0x10080000L) |
| CP_OR_SNA_SVCMG_UNDELETABLE | (0xF3010000L) |
| CPSVCMG_ALREADY_DEFD | (0x21020000L) |
| DEALLOC_BAD_TYPE | (0x51000000L) |
| DEALLOC_CONFIRM_BAD_STATE | (0x53000000L) |
| DEALLOC_FLUSH_BAD_STATE | (0x52000000L) |
| DEALLOC_LOG_LL_WRONG | (0x57000000L) |
| DEALLOC_NOT_LL_BDY | (0x55000000L) |
| DEF_PLU_INVALID_FQ_NAME | (0x74020000L) |
| DEL_MODE_DEFAULT_SPCD | (0xF4010000L) |
| DLC_ACTIVE | (0x01100000L) |
| DUPLICATE | (0x8D020000L) |
| DUPLICATE_CP_NAME | (0x02100000L) |
| DUPLICATE_DEST_ADDR | (0x03100000L) |
| DUPLICATE_TG_NUMBER | (0x15530000L) |
| DLC_DEACTIVATING | (0x86020000L) |
| ALREADY_STARTING | (0xC0010000L) |
| DUPLICATE_ADJ_NODE_ID | (0x04100000L) |
| DUPLICATE_PORT | (0x10100000L) |
| DUPLICATE_PORT_NUMBER | (0x05100000L) |
| DUPLICATE_PORT_NAME | (0x06100000L) |
| FLUSH_NOT_SEND_STATE | (0x61000000L) |
| INVALID_AUTO_ACT_SUPP | (0xB5020000L) |
| INVALID_CN_NAME | (0x21080000L) |
| INVALID_CNOS_SLIM | (0x17020000L) |
| INVALID_COS_SNASVCMG_MODE | (0x1C020000L) |
| INVALID_CP_NAME | (0xCA010000L) |
| INVALID_DATA_TYPE | (0x05070000L) |

Table 145 APPN Secondary Return Sense Codes (continued)

| Sense Codes | Hex |
|-----------------------------|---------------|
| INVALID_DEFAULT_RU_SIZE | (0x1D020000L) |
| INVALID_DLC | (0x10050000L) |
| INVALID_DLC_NAME | (0x07100000L) |
| INVALID_DLC_TYPE | (0x08100000L) |
| INVALID_FQ_LU_NAME | (0xFD010000L) |
| INVALID_FQ_OWNING_CP_NAME | (0xDB020000L) |
| INVALID_LIMITED_RESOURCE | (0xCE010000L) |
| INVALID_LINK_ACTIVE_LIMIT | (0x09100000L) |
| INVALID_LINK_NAME | (0xC1010000L) |
| INVALID_LINK_NAME_SPECIFIED | (0xB0020000L) |
| INVALID_LU_ALIAS | (0xB1020000L) |
| INVALID_MAX_NEGOT_SESS_LIM | (0x14020000L) |
| INVALID_MIN_CONWINNERS | (0x1E020000L) |
| INVALID_MODE_NAME | (0x15020000L) |
| INVALID_NAME_LEN | (0xC5020000L) |
| INVALID_NETID_LEN | (0xC6020000L) |
| INVALID_NODE_TYPE | (0xC4020000L) |
| INVALID_NUM_LS_SPECIFIED | (0xB2020000L) |
| INVALID_NUM_PORTS_SPECIFIED | (0x0B100000L) |
| INVALID_NUMBER_OF_NODE_ROWS | (0x02080000L) |
| INVALID_NUMBER_OF_TG_ROWS | (0x09080000L) |
| INVALID_PORT_NAME | (0x0C100000L) |
| INVALID_PORT_TYPE | (0x0D100000L) |
| INVALID_RECV_PACING_WINDOW | (0x16020000L) |
| INVALID_TARGET_PACING_CNT | (0x18020000L) |
| INVALID_TG_CHARS | (0x18030000L) |
| INVALID_TG_NUMBER | (0x15500000L) |
| INVALID_MAX_RU_SIZE_UPPER | (0x19020000L) |
| INVALID_SET_PROT | (0x00070000L) |
| INVALID_NEW_PROT | (0x01070000L) |
| INVALID_SET_UNPROT | (0x02070000L) |
| INVALID_NEW_UNPROT | (0x03070000L) |
| INVALID_SET_USER | (0x04070000L) |
| INVALID_SNASVCMG_MODE_LIMIT | (0x1A020000L) |
| INVALID_UNINT_PLU_NAME | (0x7C020000L) |
| INVALID_WILDCARD_NAME | (0x8C020000L) |
| INVALID_STATS_TYPE | (0x06070000L) |
| INVALID_TABLE_TYPE | (0x07070000L) |
| LINK_ACT_BY_LOCAL | (0x15100000L) |
| LINK_ACT_BY_REMOTE | (0x14100000L) |
| LINK_DEACTIVATED | (0x13100000L) |
| LINK_DEACT_IN_PROGRESS | (0x12100000L) |

Table 145 APPN Secondary Return Sense Codes (continued)

| Sense Codes | Hex |
|-----------------------------|---------------|
| LINK_NOT_DEFD | (0x17100000L) |
| LOCAL_CP_NAME | (0xD7010000L) |
| LS_ACTIVE | (0xDA010000L) |
| MISSING_CP_NAME | (0x15510000L) |
| MISSING_CP_TYPE | (0x15520000L) |
| MISSING_TG_NUMBER | (0x15550000L) |
| MODE_NAME_NOT_DEFD | (0xF5010000L) |
| MODE_SESS_LIM_EXCEEDS_NEG | (0x20020000L) |
| MODE_UNDELETABLE | (0xF6010000L) |
| NO_PORTS_DEFINED_ON_DLC | (0x0F100000L) |
| NO_USE_OF_SNASVCMG | (0x17000000L) |
| NO_USE_OF_SNASVCMG_CPSVCMG | (0x17000000L) |
| NODE_ROW_WGT_LESS_THAN_LAST | (0x04080000L) |
| PARALLEL_TGS_NOT_ALLOWED | (0x15570000L) |
| PIP_LEN_INCORRECT | (0x16000000L) |
| PORT_ACTIVE | (0x0E100000L) |
| PORT_DEACTIVATED | (0x08070000L) |
| PS_CREATION_FAILURE | (0x18100000L) |
| P_TO_R_INVALID_TYPE | (0xA1000000L) |
| P_TO_R_NOT_LL_BDY | (0xA2000000L) |
| P_TO_R_NOT_SEND_STATE | (0xA3000000L) |
| RCV_AND_POST_BAD_FILL | (0xD5000000L) |
| RCV_AND_POST_BAD_STATE | (0xD1000000L) |
| RCV_AND_POST_NOT_LL_BDY | (0xD2000000L) |
| RCV_AND_WAIT_BAD_FILL | (0xB5000000L) |
| RCV_AND_WAIT_BAD_STATE | (0xB1000000L) |
| RCV_AND_WAIT_NOT_LL_BDY | (0xB2000000L) |
| RCV_IMMD_BAD_FILL | (0xC4000000L) |
| RCV_IMMD_BAD_STATE | (0xC1000000L) |
| R_T_S_BAD_STATE | (0xE1000000L) |
| SECURITY_NOT_VALID | (0x5160F08L) |
| SEND_DATA_CONFIRM_SYNC_NONE | (0xF5000000L) |
| SEND_DATA_INVALID_TYPE | (0xF4000000L) |
| SEND_DATA_NOT_LL_BDY | (0xF6000000L) |
| SEND_DATA_NOT_SEND_STATE | (0xF2000000L) |
| SEND_ERROR_BAD_TYPE | (0x03010000L) |
| SEND_ERROR_LOG_LL_WRONG | (0x02010000L) |
| SNA_DEFD_COS_CANT_BE_CHANGE | (0x0A080000L) |
| SNA_DEFD_COS_CANT_BE_DELETE | (0x11080000L) |
| STOP_PORT_PENDING | (0x11100000L) |
| TG_NUMBER_IN_USE | (0x15540000L) |
| TG_ROW_WGT_LESS_THAN_LAST | (0x05080000L) |

Table 145 APPN Secondary Return Sense Codes (continued)

| Sense Codes | Hex |
|-----------------------------|---------------|
| TRANS_PGM_NOT_AVAIL_NO_RTRY | (0x00004C08L) |
| TRANS_PGM_NOT_AVAIL_RETRY | (0x31604B08L) |
| TP_NAME_NOT_RECOGNIZED | (0x21600810L) |
| UNKNOWN_PARTNER_MODE | (0x18000000L) |
| UNRECOGNIZED_DEACT_TYPE | (0x0E050000L) |
| LU_NAME_WILDCARD_NAME_CLAH | (0x8E020000L) |
| TP_ACTIVE | (0x19100000L) |
| MODE_ACTIVE | (0x1A100000L) |
| PLU_ACTIVE | (0x1B100000L) |
| INVALID_PLU_NAME | (0x1C100000L) |
| INVALID_SET_NEGOTIABLE | (0x1D100000L) |
| INVALID_MODE_NAME_SELECT | (0x1E100000L) |
| INVALID_RESPONSIBLE | (0x1F100000L) |
| INVALID_DRAIN_SOURCE | (0x20100000L) |
| INVALID_DRAIN_TARGET | (0x21100000L) |
| INVALID_FORCE | (0x22100000L) |
| INVALID_CLEANUP_TYPE | (0x24100000L) |
| INVALID_COS_NAME | (0x25100000L) |
| INVALID_SESSION_LIMIT | (0x26100000L) |
| INVALID_DRAIN | (0x27100000L) |
| INVALID_PRL_SESS_SUPP | (0x28100000L) |
| INVALID_LU_NAME | (0x29100000L) |
| MODE_NOT_RESET | (0x2A100000L) |
| MODE_RESET | (0x2B100000L) |
| CNOS_REJECT | (0x2C100000L) |
| CNOS_COMMAND_RACE_REJECT | (0x5F010000L) |
| CNOS_MODE_NAME_REJECT | (0x57010000L) |
| INVALID_OP_CODE | (0x2D100000L) |
| EXCEEDS_MAX_ALLOWED | (0x5C010000L) |
| DEACT_CG_INVALID_CGID | (0x6C020000L) |
| INVALID_SESSION_ID | (0x12050000L) |
| LU_NAU_ADDR_ALREADY_DEFD | (0X12020000L) |
| DIR_ENTRY_PARENT | (0x38100000L) |
| NODE_ALREADY_STARTED | (0xZ3910000L) |
| NODE_FAILED_TO_START | (0x3A100000L) |
| LU_ALREADY_DEFINED | (0x3B100000L) |
| PORT_INACTIVE | (0x3D100000L) |
| ACTIVATION_LIMITS_REACHED | (0x3E100000L) |
| PARALLEL_TGS_NOT_SUPPORTED | (0x3F100000L) |
| DLC_INACTIVE | (0x40100000L) |
| NO_LINKS_DEFINED | (0x41100000L) |
| STOP_DLC_PENDING | (0x42100000L) |

Table 145 APPN Secondary Return Sense Codes (continued)

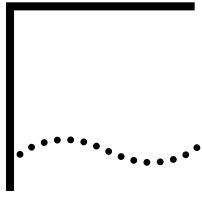
| Sense Codes | Hex |
|----------------------------|---------------|
| INVALID_LS_ROLE | (0x43100000L) |
| INVALID_BTU_SIZE | (0x44100000L) |
| LAST_LINK_ON_ACTIVE_PORT | (0x45100000L) |
| DYNAMIC_LOAD_ALREADY_REGD | (0x46100000L) |
| INVALID_LIST_OPTION | (0x47100000L) |
| INVALID_RES_NAME | (0x48100000L) |
| INVALID_RES_TYPE | (0x49100000L) |
| INVALID_ADJ_NNCP_NAME | (0x4A100000L) |
| INVALID_NODE | (0x4B100000L) |
| INVALID_ORIGIN_NODE | (0x4C100000L) |
| INVALID_TG | (0x4D100000L) |
| INVALID_FQPCID | (0x4E100000L) |
| INVALID_POOL_NAME | 0x4F1000000L) |
| INVALID_NAU_ADDRESS | (0x50100000L) |
| INVALID_ENABLE_POOL | (0x50300000L) |
| LU_NAME_POOL_NAME_CLASH | (0x51100000L) |
| INVALID_PRIORITY | (0x52100000L) |
| INVALID_DNST_LU_NAME | (0x53100000L) |
| INVALID_HOST_LU_NAME | (0x54100000L) |
| PU_NOT_DEFINED | (0x55100000L) |
| INVALID_PU_NAME | (0x56100000L) |
| INVALID_MAX_IFRM_RCVD | (0x57100000L) |
| INVALID_SYM_DEST_NAME | (0x58100000L) |
| INVALID_LENGTH | (0x59100000L) |
| INVALID_ISR_THRESHOLDS | (0x5A100000L) |
| INVALID_NUM_LUS | (0x5B100000L) |
| CANT_DELETE_ADJ_ENDNODE | (0x5C100000L) |
| INVALID_RESOURCE_TYPE | (0x5D100000L) |
| PU_CONC_NOT_SUPPORTED | (0x5E100000L) |
| DLUR_NOT_SUPPORTED | (0x5F100000L) |
| INVALID_RTP_CONNECTION | (0x60100000L) |
| PATH_SWITCH_IN_PROGRESS | (0x61100000L) |
| HPR_NOT_SUPPORTED | (0x62100000L) |
| RTP_NOT_SUPPORTED | (0x63100000L) |
| COS_TABLE_FULL | (0x64100000L) |
| INVALID_DAYS_LEFT | (0x65100000L) |
| CONVERSATION_TYPE_MISMATCH | (0x34600810L) |
| PIP_NOT_ALLOWED | (0x31600810L) |
| SYNC_LEVEL_NOT_SUPPORTED | (0x41600810L) |
| PLU_ALIAS_CANT_BE_CHANGED | (0xB3020000L) |
| PLU_ALIAS_ALREADY_USED | (0xB4020000L) |
| LU_ALIAS_CANT_BE_CHANGED | (0xB8020000L) |

Table 145 APPN Secondary Return Sense Codes (continued)

| Sense Codes | Hex |
|----------------------------|---------------|
| LU_ALIAS_ALREADY_USED | (0xB9020000L) |
| UNKNOWN_USER | (0x32100000L) |
| NO_PROFILES | (0x33100000L) |
| TOO_MANY_PROFILES | (0x36100000L) |
| INVALID_UPDATE_TYPE | (0x37100000L) |
| INVALID_USERID | (0x90020000L) |
| INVALID_PASSWORD | (0x91020000L) |
| INVALID_PROFILE | (0x93020000L) |
| INVALID_DLUS_NAME | (0x00900000L) |
| NO_DEFAULT_DLUS_DEFINED | (0x01900000L) |
| INVALID_PU_ID | (0x02900000L) |
| PU_ALREADY_ACTIVATING | (0x03900000L) |
| PU_ALREADY_DEACTIVATING | (0x04900000L) |
| PU_ALREADY_ACTIVE | (0x05900000L) |
| PU_NOT_ACTIVE | (0x06900000L) |
| DLUS_REJECTED | (0x07900000L) |
| DLUS_CAPS_MISMATCH | (0x08900000L) |
| PU_FAILED_ACTPU | (0x09900000L) |
| PU_NOT_RESET | (0x0A900000L) |
| PU_OWNS_LUS | (0x0B900000L) |
| INVALID_FILTER_OPTION | (0x0C900000L) |
| INVALID_STOP_TYPE | (0x0D900000L) |
| PU_ALREADY_DEFINED | (0x0E900000L) |
| DEPENDENT_LU_NOT_SUPPORTED | (0x0F900000L) |
| INVALID_DSPU_NAME | (0x12900000L) |
| DSPU_ALREADY_DEFINED | (0x13900000L) |
| INVALID_SOLICT_SSCP_SESS | (0x14900000L) |
| INVALID_BACK_LEVEL_SUPPORT | (0x15000000L) |
| INVALID_BKUP_DLUS_NAME | (0x15900000L) |
| INVALID_EFFECTIVE_CAPACITY | (0x24080000L) |
| INVALID_TIME_COST | (0xD6010000L) |
| INVALID_TP_NAME | (0xA0020000L) |
| INVALID_BYTE_COST | (0xD1010000L) |
| DEF_LINK_INVALID_SECURITY | (0x22080000L) |
| INVALID_PROPAGATION_DELAY | (0x23080000L) |
| INVALID_USER_DEF_1 | (0xC3010000L) |
| INVALID_USER_DEF_2 | (0xC4010000L) |
| INVALID_USER_DEF_3 | (0xC5010000L) |
| AS_NEGOTIATED | (0x07000000L) |
| AS_SPECIFIED | (0x00000000L) |
| FORCED | (0xB7020000L) |
| INVALID_LS_NAME | (0xB7030000L) |

Table 145 APPN Secondary Return Sense Codes (continued)

| Sense Codes | Hex |
|---------------------------|---------------|
| INVALID_LFSID_SPECIFIED | (0xB7040000L) |
| INVALID_FILTER_TYPE | (0xB7050000L) |
| INVALID_MESSAGE_TYPE | (0xB7060000L) |
| CANT_DELETE_CP_LU | (0xB7070000L) |
| ALL_RESOURCES_NOT_DEFINED | (0xB7090000L) |
| INVALID_LIST_TYPE | (0xB70A0000L) |



IBM TRACE FACILITY

This appendix describes how to set up filters to capture traces of data link switching (DLSw), Logical Link Control type 2 (LLC2), or Synchronous Data Link Control (SDLC) packets to troubleshoot IBM network environment problems. The trace facility uses mnemonic filtering masks to filter specific types of packets for tracing purposes. For more information about mnemonic filtering, see the *Configuring Mnemonic Filtering* chapter. For more information about parameters in the Filter Service, see the *Filter Service Parameters* chapter in *Reference for Enterprise OS Software*.

Tracing IBM Data Traffic

You can trace IBM data traffic of the following packet frame types:

- DLSw
- LLC2
- SDLC

This appendix is divided into sections showing how to trace each packet frame type.



In the examples in this chapter, all MAC addresses must be entered in noncanonical format.

Tracing DLSw Packets

To set up a trace for DLSw packets, you set up mnemonic filters and masks, follow these steps:

1 Set up the mask using:

```
ADD -Filter MASK <maskname> <location> <pattern>
<location>:= <mnemonic format>
<mnemonic format>:= <protocol>.<field> <protocol>:= DLSW <field>:=
DLSwLclMAC | DLSwLclSAP | DLSwRmtMAC |
DLSwRmtSAP | IPADDRESS <maskname> is an arbitrary string of 15 printable
characters
```

You can set up the field in the mask in several ways to trace specific types of packets from the following locations:

- DLSw local MAC address
- Remote MAC address
- DLSw local SAP
- DLSw remote SAP
- A specific IP address

Table 146 lists the possible fields and the appropriate matching value. For examples of how to set up these types of masks, see “DLSw Filter Examples” later in this chapter.

Table 146 Field Values for DLSw Traces

| Field | Description | Matching Value |
|------------|-------------------------------|---------------------|
| DlswLcIMAC | Local MAC address | <MAC address> |
| DlswLcISAP | Local SAP | <hexadecimal value> |
| DlswRmtMAC | Remote MAC address | <MAC address> |
| DlswRmtSAP | Remote SAP | <hexadecimal value> |
| IPADDRess | IP address of the DLSw tunnel | <IP address> |

You can display these values by entering:

```
SHOW -Filter MNemonics DLSw
```

- 2 Set up the filter policy using:

```
ADD -Filter POLicy <policyname><action> <masks>
```

Specify the action as TRace.

For the <masks> value, you can select one of two built-in masks and/or masks you have defined. Table 147 lists the built-in masks for tracing different types of packets.

Table 147 Built-in Masks for Tracing DLSw Packets

| Built-in Mask | Equivalent | Packet Type |
|---------------|------------|--------------------------|
| DLCTL | DLSW.1=72 | DLSw Control Message |
| DLSWI | DLSW.1=16 | DLSw Information Message |

For examples of how to set specific DLSw masks and policies, see “DLSw Filter Examples” later in this chapter.



When setting policies for DLSw, the only action allowed is TRace.

- 3 Set the maximum number of bytes to be captured in the trace using:

```
SETDefault -DLSw MaxTRaceData = <max_bytes_captured> (0-76)
```

This parameter sets the number of bytes captured over and above the DLSw message headers. The number of bytes you capture determines the quality of the trace data. The more bytes you capture, the more information you will receive. The number you specify will be rounded up to the nearest multiple of four when determining how many bytes to capture. For example, if you set the value to 29, the maximum number of bytes to be captured is rounded up to 32.

- 4 Set the filter selection by entering:

```
SETDefault Filter SElection = DLSW
```

- 5 Enable the Filter Service by entering one of the following commands:

```
SETDefault -Filter CONTrol = (Enabled, MatchOne)
```

or

```
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

For more information about parameters in the Filter Service, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Displaying DLSw Trace Data To display the trace data, enter:

```
SHoW -DLsw TRaceData
```

DLSw Filter Examples This section provides examples for setting up different filters for tracing DLSw packets.

Example 1 **Tracing DLSw Packets from a Local MAC Address**

To trace DLSw packets from DLSw local MAC address %600631244F6F with a mask named DLSW1 and policy name EX1, enter:

```
ADD -Filter MASK DLSW1 DLSW.DlswLclMac = %00631244F6F
ADD -Filter POLicy EX1 TRace DLSW1
```

Example 2 **Tracing DLSw Packets from a Local SAP**

To trace DLSw packets from DLSw local SAP %04 with a mask named DLSW2 and policy name EX2, enter:

```
ADD -Filter MASK DLSW2 DLSW.DlswLclSap = %04
ADD -Filter POLicy EX2 TRace DLSW2
```

Example 3 **Tracing DLSw Packets from a Remote MAC Address**

To trace DLSw packets from DLSw remote MAC address %6020000C0E854 with a mask named DLSW3 and policy name EX3, enter:

```
ADD -Filter MASK DLSW3 DLSW.DlswRmtMac = %020000C0E854
ADD -Filter POLicy EX3 TRace DLSW3
```

Example 4 **Tracing DLSw Packets from a Remote SAP**

To trace DLSw packets from DLSw remote SAP %04 with a mask named DLSW4 and policy name EX4, enter:

```
ADD -Filter MASK DLSW4 DLSW.DlswRmtSap = %04
ADD -Filter POLicy EX4 TRace DLSW4
```

Example 5 **Tracing DLSw Packets from an IP Address**

To trace DLSw packets from IP address 129.213.240.230 with a mask named DLSW5 and policy name EX5, enter:

```
ADD -Filter MASK DLSW5 DLSW.IPADDRESS = 129.213.240.230
ADD -Filter POLicy EX5 TRace DLSW5
```

Example 6 **Tracing DLSw Control Message Packets from a Local MAC Address**

To trace DLSw control message packets from DLSw local MAC address %600631244F6F with a mask named DLSW6 and policy name EX6, enter:

```
ADD -Filter MASK DLSW6 DLSW.DlswLclMac = %00631244F6F
ADD -Filter POLicy EX6 TRace DLSWCTL,DLSW6
```

Example 7 **Tracing DLSw Control Message Packets from a Local SAP**

To trace DLSw control message packets from DLSw local SAP %04 with a mask named DLSW7 and policy name EX7, enter:

```
ADD -Filter MASK DLSW7 DLSW.DlswLclSap = %04
ADD -Filter POLicy EX7 TRace DLSWCTL,DLSW7
```

Example 8 **Tracing DLSw Control Message Packets from a Remote MAC Address**

To trace DLSw control message packets from DLSw remote MAC address %6020000C0E854 with a mask named DLSW8 and policy name EX8, enter:

```
ADD -Filter MASK DLSW8 DLSW.DlswRmtMac = %020000C0E854
ADD -Filter POLicy EX8 TRace DLSWCTL,DLSW8
```

Example 9 **Tracing DLSw Control Message Packets from a Remote SAP**

To trace DLSw control message packets from DLSw remote SAP %04 with a mask named DLSW9 and policy name EX9, enter:

```
ADD -Filter MASK DLSW9 DLSW.DlswRmtSap = %04
ADD -Filter POLicy EX9 TRace DLSWCTL,DLSW9
```

Example 10 **Tracing DLSw Control Message Packets from an IP Address**

To trace DLSw control message packets from IP address 129.213.240.230 with a mask named DLSW10 and policy name EX10, enter:

```
ADD -Filter MASK DLSW10 DLSW.IPADDRESS = 129.213.240.230
ADD -Filter POLicy EX10 TRace DLSWCTL,DLSW10
```

Example 11 **Tracing DLSw Information Message Packets from a Local MAC Address**

To trace DLSw information message packets from DLSw local MAC address %00631244F6F with a mask named DLSW11 and policy name EX11, enter:

```
ADD -Filter MASK DLSW11 DLSW.DlswLclMac = %00631244F6F
ADD -Filter POLicy EX11 TRace DLSWI,DLSW11
```

Example 12 **Tracing DLSw Information Message Packets from a Local SAP**

To trace DLSw information message packets from DLSw local SAP %04 with a mask named DLSW12 and policy name EX12, enter:

```
ADD -Filter MASK DLSW12 DLSW.DlswLclSap = %04
ADD -Filter POLicy EX12 TRace DLSWI,DLSW12
```

Example 13 **Tracing DLSw Information Message Packets from a Remote MAC Address**

To trace DLSw information message packets from DLSw remote MAC address %6020000C0E854 with a mask named DLSW13 and policy name EX13, enter:

```
ADD -Filter MASK DLSW13 DLSW.DlswRmtMac = %020000C0E854
ADD -Filter POLicy EX13 TRace DLSWI,DLSW11
```

Example 14 **Tracing DLSw Information Message Packets from a Remote SAP**

To trace DLSw information message packets from DLSw remote SAP %04 with a mask named DLSW14 and policy name EX14, enter:

```
ADD -Filter MASK DLSW14 DLSW.DlswRmtSap = %04
ADD -Filter POLicy EX14 TRace DLSWI,DLSW14
```

Example 15 **Tracing DLSw Information Message Packets from an IP Address**

To trace DLSw information message packets from IP address 129.213.240.230 with a mask named DLSW15 and policy name EX15, enter:

```
ADD -Filter MASK DLSW15 DLSW.IPADDRESS = 129.213.240.230
ADD -Filter POLicy EX15 TRace DLSWI,DLSW15
```

Tracing LLC2 Frames To set up a trace for LLC2 frames, you set up mnemonic filters and masks, follow these steps:

1 Set up the mask using:

```
ADD -Filter MASK <maskname> <location> <pattern>
<location>:= <mnemonic format>
<mnemonic format>:= <protocol>.<field> <protocol>:= LLC2 <field>:=
FrameType | LlcLclMAC | LlcLclSAP |
LlcRmtMAC | LlcRmtSAP <pattern>:= <comparison><match>
<match>:= LlcInfoFrame | LlcUnnFrame | LlcSupFrame
```

You can set up the field in the mask in several ways to trace specific types of packets: For examples of how to set up these types of masks, see the specific examples following this section. Table 148 lists the field options available for tracing LLC2 packets from different origins and targets. For examples of how to set up these types of masks, see the specific examples in “LLC2 Filter Examples” later in this chapter.

Table 148 Field Values for LLC2 Traces

| Field | Description | Matching Value |
|-----------|--------------------|-----------------------|
| LlcLclMAC | LocalMAC address | <MAC address> |
| LlcLclSAP | Local SAP | <hexadecimal value> |
| LlcRmtMAC | Remote MAC address | <MAC address> |
| LlcRmtSAP | Remote SAP | <hexadecimal value> |
| FrameType | LLC2 frame type | <frame_type mnemonic> |

If you specify FrameType as the field value, when you set up the pattern, you set up the comparison and match. For the match, you specify the frame type mnemonic you want matched. Table 149 lists the frame type mnemonic options for tracing LLC2 packets.

Table 149 Frame Type Mnemonics for LLC2 Traces

| Frame Type Mnemonic | Equivalent | Packet Type |
|---------------------|---------------------|-------------------------|
| LlcInfoFrame | LLC2.FrameType = %0 | LLC2 information frames |
| LlcUnnFrame | LLC2.FrameType = %3 | LLC2 unnumbered frames |
| LlcSupFrame | LLC2.FrameType = %1 | LLC2 supervisory frames |

You can display these values by entering:

```
SHow -Filter MNemonics LLC
```

2 Set up the filter policy using:

```
ADD -Filter POLicy <policyname><action> <masks> <context>
```

Specify the action as TRace.

For examples of how to set specific LLC2 masks and policies, see “LLC2 Filter Examples” later in this chapter.



When setting policies for LLC2, the only action allowed is TRace.

- 3 Set the maximum number of bytes to be captured in the trace using:

```
SETDefault -LLC2 MaxTraceData = <max_bytes_captured> (0-76)
```

This parameter sets the number of bytes captured over and above the LLC2 address and control bytes. The number of bytes you capture determines the quality of the trace data. The more bytes you capture, the more information you receive. The number you specify is rounded up to the nearest multiple of four. For example, if you set the value to 29, the maximum number of bytes to be captured is rounded up to 32.

- 4 Set the filter selection by entering:

```
SETDefault -Filter SElection = LLC
```

- 5 Enable the Filter Service by entering one of the following commands:

```
SETDefault -Filter CONTrol = (Enabled, MatchOne)
```

or

```
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

For more information about parameters in the Filter Service, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Displaying LLC2 Trace Data

After you have conducted your trace, you can display the trace data by entering:

```
SHow -LLC2 TRaceData
```

LLC2 Filter Examples

This section provides examples for setting up different filters for tracing LLC2 packets.

Example 1 Tracing LLC2 Packets from a Local MAC Address

To trace LLC2 packets from local MAC address %6080002057Ab0 with a mask named LLC2_16 and policy name LLC2TRACE1, enter:

```
ADD -Filter MASK LLC2_16 LLC2.LLC2LclMac = %080002057AB0
ADD -Filter POLicy LLC2TRACE1 TRace LLC,LLC2_16
```

Example 2 Tracing LLC2 Packets from a Local SAP

To trace LLC2 packets from local SAP %08 with a mask named LLC2_17 and policy name LLC2TRACE2, enter:

```
ADD -Filter MASK LLC2_17 LLC2.LLC2LclSap = %08
ADD -Filter POLicy LLC2TRACE2 TRace LLC,LLC2_17
```

Example 3 Tracing LLC2 Packets from a Remote MAC Address

To trace LLC2 packets from remote MAC address %600608C23EBBC with a mask named LLC2_18 and policy name LLC2TRACE3, enter:

```
ADD -Filter MASK LLC2_18 LLC2.LLC2RmtMac = %00608C23EBBC
ADD -Filter POLicy LLC2TRACE3 TRace LLC,LLC2_18
```

Example 4 Tracing LLC2 Packets from a Remote SAP

To trace LLC2 packets from remote SAP %1C with a mask named LLC2_19 and policy name LLC2TRACE4, enter:

```
ADD -Filter MASK LLC2_19 LLC2.LLC2RmtSap = %1C
ADD -Filter POLicy LLC2TRACE4 TRace LLC,LLC2_19
```

Example 5 Tracing LLC2 Information Frames from a Local MAC Address

To trace LLC2 information frames from local MAC address %6080002057AB0 with masks LLC2_20 and LLC2_20A and policy name LLC2TRACE5, enter:

```
ADD -Filter MASK LLC2_20 LLC2.LLC2LclMac = %080002057AB0
ADD -Filter MASK LLC2_20A LLC2.LLC2FrameType = LlcInfoFrame
ADD -Filter POLicy LLC2TRACE5 TRace LLC2_20,LLC2_20A
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 6 Tracing LLC2 Information Frames from a Local SAP

To trace LLC2 information frames from local SAP %08 with masks LLC2_21 and LLC2_21A and policy name LLC2TRACE6, enter:

```
ADD -Filter MASK LLC2_21 LLC2.LLC2LclSap = %08
ADD -Filter MASK LLC2_21A LLC2.LLC2FrameType = LlcInfoFrame
ADD -Filter POLicy LLC2TRACE6 TRace LLC2_21,LLC2_21A
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 7 Tracing LLC2 Information Frames from a Remote MAC Address

To trace LLC2 information frames from remote MAC address %600608C23EBBC with masks LLC2_22 and LLC2_22A and policy name LLC2TRACE7, enter:

```
ADD -Filter MASK LLC2_22 LLC2.LLC2RmtMac = %080002057AB0
ADD -Filter MASK LLC2_22A LLC2.LLC2FrameType = LlcInfoFrame
ADD -Filter POLicy LLC2TRACE7 TRace LLC2_22,LLC2_22A
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 8 Tracing LLC2 Information Frames from a Remote SAP

To trace LLC2 information frames from remote SAP %1C with masks LLC2_23 and LLC2_23A and policy name LLC2TRACE8, enter:

```
ADD -Filter MASK LLC2_23 LLC2.LLC2RmtSap = %1C
ADD -Filter MASK LLC2_23A LLC2.LLC2FrameType = LlcInfoFrame
ADD -Filter POLicy LLC2TRACE8 TRace LLC2_23,LLC2_23A
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 9 Tracing LLC2 Unnumbered Frames from a Local MAC Address

To trace LLC2 unnumbered frames from local MAC address %6080002057AB0 with masks LLC2_24 and LLC2_24A and policy name LLC2TRACE9, enter:

```
ADD -Filter MASK LLC2_24 LLC2.LLC2LclMac = %080002057AB0
ADD -Filter MASK LLC2_24A LLC2.LLC2FrameType = LlcUnnFrame
ADD -Filter POLicy LLC2TRACE9 TRace LLC2_24,LLC2_24A
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 10 Tracing LLC2 Unnumbered Frames from a Local SAP

To trace LLC2 unnumbered frames from local SAP %08 with masks LLC2_25 and LLC2_25A and policy name LLC2TRACE10, enter:

```
ADD -Filter MASK LLC2_25 LLC2.LLC2LclSap = %08
ADD -Filter MASK LLC2_25A LLC2.LLC2FrameType = LlcUnnFrame
ADD -Filter POLicy LLC2TRACE10 TRace LLC2_25,LLC2_25A
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 11 Tracing LLC2 Unnumbered Frames from a Remote MAC Address

To trace LLC2 unnumbered frames from remote MAC address %600608C23EBBC with masks LLC2_26 and LLC2_26A and policy name LLC2TRACE11, enter:

```
ADD -Filter MASK LLC2_26 LLC2.LLC2RmtMac = %080002057AB0
ADD -Filter MASK LLC2_26A LLC2.LLC2FrameType = LlcUnnFrame
ADD -Filter POLicy LLC2TRACE7 TRace LLC2_26,LLC2_26A
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 12 Tracing LLC2 Unnumbered Frames from a Remote SAP

To trace LLC2 unnumbered frames from remote SAP %1C with masks LLC2_27 and LLC2_27A and policy name LLC2TRACE12, enter:

```
ADD -Filter MASK LLC2_27 LLC2.LLC2RmtSap = %1C
ADD -Filter MASK LLC2_27A LLC2.LLC2FrameType = LlcUnnFrame
ADD -Filter POLicy LLC2TRACE12 TRace LLC2_27,LLC2_27A
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Tracing SDLC Frames To set up a trace for SDLC frames, you set up mnemonic filters and masks, follow these steps:

1 Set up the mask using:

```
ADD -Filter MASK <maskname> <location> <pattern>
<location>:= <mnemonic format>
<mnemonic format>:= <protocol>.<field> <protocol>:= SDLC <field>:=
FrameType | PollADDRESS <pattern>:= <comparison>
<match> <match>:= SDLCInfoFrame | SDLCUnnFrame | SDLCSupFrame
```

You can set up the field in the mask in several ways to trace specific types of packets. For examples of how to set up these types of masks, see the specific examples following this section. Table 150 lists the field options available for tracing SDLC packets from different origins and targets. For examples of how to set up these types of masks, see the specific examples in “SDLC Filter Examples” later in this chapter.

Table 150 Field Values for SDLC Traces

| Field | Description | Matching Value |
|-------------|-------------------|-----------------------|
| FrameType | SDLC frame type | <frame_type mnemonic> |
| PollADDRESS | SDLC Poll Address | <hexadecimal value> |

If you specify FrameType as the field value, when you set up the pattern, you set up the comparison and match. For the match, you specify the frame type mnemonic you want matched. Table 151 lists the frame type mnemonic options for tracing LLC2 packets.

Table 151 Frame Type Mnemonics for SDLC Traces

| Frame Type Mnemonic | Equivalent | Packet Type |
|---------------------|---------------------|------------------------|
| SDLCInfoFrame | SDLC.FrameType = %0 | SDLC Info frames |
| SDLCUnnFrame | SDLC.FrameType = %3 | SDLC unnumbered frames |
| SDLCSupFrame | SDLC.FrameType = %1 | SDLC supervisor frames |

You can display these values by entering:

```
SHow -Filter Mnemonics SDLC
```

2 Set up the filter policy using:

```
ADD -Filter POLicy <polycyname><action> <masks> <context>
```

Specify the action as TRace.

For examples of how to set specific SDLC masks and policies, see “SDLC Filter Examples” later in this chapter.

3 Set the maximum number of bytes to be captured in the trace using:

```
SETDefault -SDLC MaxTRaceData = <max_bytes_captured> (0-76)
```

This parameter sets the number of bytes captured over and above the SDLC address and control bytes. The number of bytes you capture determines the quality of the trace data. The more bytes you capture, the more information you receive. The number you specify is rounded up to the nearest multiple of four. For example, if you set the value to 29, the maximum number of bytes to be captured is rounded up to 32.

4 Set the filter selection by entering:

```
SETDefault -Filter SElection = SDLC
```

5 Enable the Filter Service by entering one of the following commands:

```
SETDefault -Filter CONTrol = (Enabled, MatchOne)
```

or

```
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

For more information about parameters in the Filter Service, see the Filter Service Parameters chapter in *Reference for Enterprise OS Software*.

Displaying SDLC Trace Data

After you have conducted your trace, you can display the trace data by entering:

```
SHow -SDLC TraceData
```

SDLC Filter Examples

This section provides examples for setting up different filters for tracing SDLC packets.

Example 1 **Tracing SDLC Packets from a Poll Address**

To trace SDLC packets from poll address %C1 with mask SDLC1 and policy SDLCTRACE1 on port 2, enter:

```
ADD -Filter MASK SDLC1 SDLC.PollADDRESS = %C1
ADD -Filter POLicy SDLCTRACE1 TRace SDLC1 at !2
```

Example 2 **Tracing SDLC Information Frames**

To trace SDLC information frames from poll address %C1 with masks SDLC1 and SDLC2 and policy SDLCTRACE2 on port 2, enter:

```
ADD -Filter MASK SDLC1 SDLC.PollADDRESS = %C1
ADD -Filter MASK SDLC2 SDLC.FrameType = SDLCInfoFrame
ADD -Filter POLicy SDLCTRACE2 TRace SDLC1,SDLC2 at !2
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 3 **Tracing SDLC Unnumbered Frames**

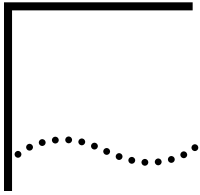
To trace SDLC unnumbered frames from poll address %C1 with masks SDLC1 and SDLC3 and policy SDLCTRACE3 on port 2, enter:

```
ADD -Filter MASK SDLC1 SDLC.PollADDRESS = %C1
ADD -Filter MASK SDLC3 SDLC.FrameType = SDLCUnnFrame
ADD -Filter POLicy SDLCTRACE3 TRace SDLC1,SDLC3 at !2
SETDefault -Filter CONTrol = (Enabled, CheckAll)
```

Example 4 **Tracing SDLC Unnumbered Frames**

To trace SDLC supervisory frames from poll address %C1 with masks SDLC1 and SDLC4 and policy SDLCTRACE4 on port 2, enter:

```
ADD -Filter MASK SDLC1 SDLC.PollAddress = %C1
ADD -Filter MASK SDLC4 SDLC.FrameType = SDLCSupFrame
ADD -Filter POLicy SDLCTRACE4 TRace SDLC1,SDLC4 at !2
SETDefault -Filter CONTROL = (Enabled, CheckAll)
```

DLSw, APPN, AND BSC HOST CONFIGURATION EXAMPLES

This appendix provides examples of how to configure hosts or front end processors to map to NETBuilder bridge/router DLSw, APPN, and BSC configurations. It also provides information on issuing NetView run commands to access the NETBuilder bridge/router from NetView.



All host displays in this chapter are copyright, IBM Corporation.

DLSw Host Examples

This section provides sample host configurations for configuring DLSw between a host and a NETBuilder bridge/router.

Example 1: Configuring a 3745 Host with Dual TIC to Support BAN

This example shows how to configure a host to support BAN Frame Relay to a NETBuilder bridge/router.

The following is the configuration required on the 3745 host:

```

*****
* MEMBER CREATED FOR DUAL TIC 3745 TESTING *
*****
SWFRBAN1 VBUILD TYPE=SWNET,MAXGRP=5,MAXNO=5
*
PUFRB91
        PUADDR=01,
        DISCNT=NO,
        IDBLK=05D,
        IDNUM=B9001,
        MAXDATA=512,
        MAXOUT=7,
        MAXPATH=2,
        DLOGMOD=SNX32702,
        PACING=0,
        PASSLIM=7,
        PUTYPE=2,
        SSCPFM=USSSCS,
        USSTAB=USSTEST,
        VPACING=0
        LU FR9102LULOCADDR=2
LUF9103 LULOCADDR=3
LUF9104 LULOCADDR=4
LUF9105 LULOCADDR=5
*
PUFRB92
        PUADDR=01,
        DISCNT=NO,
        IDBLK=05D
        IDNUM=B9002,
        MAXDATA=512,
        MAXOUT=7,
        MAXPATH=1,

```

| | | |
|---------|-------------------|-----------|
| | DLOGMOD=SNX32702, | X00320000 |
| | PACING=0, | X00330000 |
| | PASSLIM=7, | X00340000 |
| | PUTYPE=2, | X00350000 |
| | SSCPFM=USSSCS, | X00360000 |
| | USSTAB=USSTEST, | X00370000 |
| | VPACING=0 | 00380000 |
| LUF9202 | LULOCADDR=2 | 00390000 |
| LUF9203 | LULOCADDR=3 | 00400000 |
| LUF9204 | LULOCADDR=4 | 00410000 |
| LUF9205 | LULOCADDR=5 | 00420000 |
| * | | 00430000 |
| PUFRB93 | PUADDR=01, | X00440000 |
| | DISCNT=NO, | X00450000 |
| | IDBLK=05D, | X00460000 |
| | IDNUM=B9003, | X00470000 |
| | MAXDATA=512, | X00480000 |
| | MAXOUT=7, | X00490000 |
| | MAXPATH=1, | X00500000 |
| | DLOGMOD=SNX32702, | X00510000 |
| | PACING=0, | X00520000 |
| | PASSLIM=7, | X00530000 |
| | PUTYPE=2 | X00540000 |
| | SSCPFM=USSSCS, | X00550000 |
| | USSTAB=USSTEST, | X00560000 |
| | VPACING=0 | 00570000 |
| LUF9302 | LU LOCADDR=2 | 00580000 |
| LUF9303 | LU LOCADDR=3 | 00590000 |
| LUF9304 | LU LOCADDR=4 | 00600000 |
| LUF9305 | LU | 00610000 |
| * | | 00620000 |
| PUFRB94 | PUADDR=01, | X00630000 |
| | DISCNT=NO, | X00640000 |
| | IDBLK=05D, | X00650000 |
| | IDNUM=B9004, | X00660000 |
| | MAXDATA=512, | X00670000 |
| | MAXOUT=7, | X00680000 |
| | MAXPATH=1, | X00690000 |
| | DLOGMOD=SNX32702, | X00700000 |
| | PACING=0, | X00710000 |
| | PASSLIM=7, | X00720000 |
| | PUTYPE=2, | X00730000 |
| | SSCPFM=USSSCS, | X00740000 |
| | USSTAB=USSTEST, | X00750000 |
| | VPACING=0 | 00760000 |
| LUF9402 | LU LOCADDR=2 | 00770000 |
| LUF9403 | LU LOCADDR=3 | 00780000 |
| LUF9404 | LU LOCADDR=4 | 00790000 |
| LUF9405 | LULOCADDR=5 | 00800000 |
| * | | 00810000 |
| PUFRB95 | PUADDR=01, | X00820000 |
| | DISCNT=NO, | X00830000 |
| | IDBLK=05D, | X00840000 |
| | IDNUM=B9005, | X00850000 |
| | MAXDATA=512, | X00860000 |
| | MAXOUT=7, | X00870000 |
| | MAXPATH=1, | X00880000 |
| | DLOGMOD=SNX32702, | X00890000 |
| | PACING=0, | X00900000 |
| | PASSLIM=7, | X00910000 |

```

                PUTYPE=2,                X00920000
                SSCPFM=USSSCS,          X00930000
                USSTAB=USSTEST,         X00940000
                VPACING=0                00950000
LUF9592        LU LOCADDR=2            00960000
LUF9593        LU LOCADDR=3            00970000
LUF9594        LU LOCADDR=4            00980000
LUF9595        LU LOCADDR=5            00990000
*                                                    01000000
    
```

Example 2: Configuring a Host to Support Boundary Access Node (BAN) Frame Relay Between a Host and a NETBuilder Bridge/Router

This example shows how to configure a host to support BAN Frame Relay directly to a NETBuilder bridge/router.

The following is the configuration on the NETBuilder bridge/router:

```

===== -DLSW BoundAccessNode parameter menu (Level 3)=====
 1 - Add
 2 - Delete
Select (1-2) ... <CR> to Exit ==== 1
Enter !<port> (mandatory) => 1
ADD !<Vport> BoundAccessNode <ban dlci mac addr> [<bni mac addr>]
ADD !V1 BoundAccessNode 4FFF00000000
    
```

The following is the configuration required on the host (entries underlined in the host example map directly to the configuration required on the NETBuilder bridge/router):

```

*****                                                    00010000
*   SWITCHED MAJOR NODE FOR BAN FRAME RELAY   *           00020000
*****                                                    00030000
**                                                    **           00040000
SNAFRBAN      VBUILD TYPE=SWNET,MAXGRP=1,MAXNO=6          00050000
*                                                    00060000
PUFRBAN1      PUADDR=01, IDBLK=05D,IDNUM=00099, 21       X00070000
                MAXPATH=3,MAXDATA=1024,PUTYPE=2          X00080003
                IRETRY=NO,DISCNT=NO,ISTATUS=ACTIVE       X00090000
                MAXOUT=7,PASSLIM=7                       X00100000
                USSTAB=USSTEST                            00110000
BANPTH1       PATHDLCADDR=(1,C,FRELAY),DLC TYPE IS FRAME-RELAY X00120002
                DLCADDR=(2,D,02), PORT#OF PHY LN (PORTADD) X00130006
                DLCADDR=(3,D,8) SAP OF FRAME-RELAY DEVICE X00140001
                DLCADDR=(5,X,4FFF00000000),DEST MAC ADDR FOR BAN X00160002
                GID=1,PID=1                                X00170000
                GRPN=G1-FRLG0                              00180000
BANPTH2       PATHDLC ADDR+(1,C,FRELAY,DLC TYPE IS FRAME-RELAY X00190002
                DLCADDR=(2,D,03), PORT#OF PHYS LN (PORTADD) X00200006
                DLCADDR=(3,D,8), SAP OF FRAME-RELAY DEVICE X00210002
                DLCADDR=(4,X,20), DLCI OF FRAME-RELAY PVC X00220002
                DLCADDR=(5,X,4FFF00000000),DEST MAC ADDR FOR BAN X00230002
                GID=1.PID=2                                X00240005
                GRPNM=G10FRLG0                            00320002
    
```

```

**A0488L21 LU LOCADDR=00,DLOGMOD=DSIL6MOD,MODETAB=AMODETAB      00330000
LUFRBA11 LU LOCADDR=01,DLOGMOD=SNX32702                          00340000
LUFRBA12 LU LOCADDR=02,DLOGMOD=SNZ32702                          00350000
LUFRBA13 LU LOCADDR=03,DLOGMOD=SNX32702                          00360000
LUFRBA14 LU LOCADDR=04,DLOGMOD=SNX32702                          00370000
*                                                                    00380000

```

Note the following about this example:

- The values you enter on the bridge/router for the <fep mac> <fep sap> syntax come from the DLCADDR parameter in the switched major node for the frame relay connection. This defines the token ring interface on the FEP.

Example 3: Configuring a Host to Support Boundary Network Node (BNN) Frame Relay Between a Host and a NETBuilder Bridge/Router

This example shows how to configure a host to support BNN Frame Relay to a NETBuilder bridge/router.

The following is the configuration on the NETBuilder bridge/router:

```

===== SHow -DLSW FradMap =====
No FradMap Configured
===== -DLSW FradMap parameter menu (Level 3)=====
  1 - Add
  2 - Delete
Select (1-2) ... <CR> to Exit =====> 1
Enter !<port> (mandatory) => 1
  Add !<port> FradMap <src mac> <src sap> <fep mac> <fep sap> <dlci>
  <code point>
  Add !1 FradMap 4FFF00000000 04 400011600000 04 10 82

```

The following is the configuration required on the host (entries underlined in the host example map directly to the configuration required on the NETBuilder bridge/router):

```

***** 00010000
* 6/14/96 MEMBER FOR FRAME RELAY BNN ON L1020 OF N10 * 00020000
* IT'S USED TO CONNECT OS/2 AT 4TH FLOOR 00021000
***** 00030000
SWFRFRAD VBUILD TYPE=SWNET,MAXGRP=2,MAXNO=2 00040000
* 00050000
PUFRF05F PU ADDR=01, X00060000
          DISCNT=NO, X00070000
          IDBLK=05D, X00080001
          IDNUM=B005F, X00090000
          MAXDATA=512, X00100000
          MAXOUT=7, X00110000
          MAXPATH=2, X00120000
          DLOGMOD=SNX32702, X00130000
          PACING=0, X00140000
          PASSLIM=7, X00150000
          PUTYPE=2, X00160000

```

```

                SSCPFM=USSSCS,                                X00170000
                USSTAB=USSTEST,                              X00180000
                VPACING=0                                    00190000
LUFRO5F2      LU LOCADDR=2                                  00193000
LUFRO5F3      LU LOCADDR=3                                  00194000
LUFRO5F4      LU LOCADDR=4                                  00195000
LUFRO5F5      LU LOCADDR=5                                  00196000
*                                                       00197000
*TFRF05F PATH DLCADDR=(1,C,FRELAY),                        X00197105
*               DLCADDR=(2,D,02),PORTADDR ON PHYSICAL LINE X00197205
*               DLCADDR=(3,D,4), REMOTE SAP                 X00197305
*               DLCADDR=(4,X,99),DLCI #                     X00197405
*               GID=1,PID=1,                                 X00197505
*               GRPNM=G10FRLG1                               00197605
* ADD NEXT LINE FOR BAN TYPE1 TO BE INITIATED BY HOST *    00197702
* TFRF05F PATH DIALNO=02040020AF00B3C1,GRPNM=G10FRLG1    00197804
*****                                                    00197902
*               SWITCHED MAJOR NODE FOR FRAME RELAY        *    00198002
*****                                                    00199002

```

APPN Host Configurations

This section provides examples showing how to configure APPN with hosts in certain situations.

Example 4: Defining an Adjacent Link Station for a TIC to a Host

This example shows how you would define an adjacent link station for a token ring interface card (TIC) connection to a host in the APPN service. The interface in this case is for a 3745 front end processor (FEP).

```

===== SHoW -APPN AdjLinkSta =====
-----Adjacent Link Stations-----
Port Linkname BTU type  Media addr  SAP  CPName          ID  TG
prof CAHE
!0 LINK0000 2048 NN n100040607FF8 04 US3COMHQ.APPN1 00000000 CAHE
!0 LINK0001 2048 NN n100040080EA3 04 US3COMHQ.APPN4 00000000 CAHE
===== -APPN AdjLinkSta parameter menu (Level 3)=====
  1 - Add
  2 - Delete
Select (1-2) ... <CR> to Exit =====> 1
Enter !<port> (mandatory) => 1
  Add      !<port> AdjLinkSta  <type>(NN|EN|Learn)
<max_btu_size>(99-8912)
  [[Cmac|Ncmac] dest media addr] [Sap=<num>]
  [CPName=<[netid.]cpname>] [Nodeid=<ID>] [LinkName=<name>]
  [TGprof=<name>] [CPSess=(Yes|No)] [AutoStart=(Yes|No)]
  [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]

Add !1 AdjLinkSta NN 2048 N100040607FF8 Sap=04 CPName=US3COMHQ.APPN1
CPSess=Yes AutoStart=Yes HPR=Yes ErrorRecovery=No

```



```

NUMHSAS=6,          NUMBER OF VR'S ENDING IN THIS NCP      *
PATHEXT=12,        EXTRA TRANSIT ROUTING TABLE ENTRIES  *
SESSACC=NO,        NO SESSION ACCOUNTING BECAUSE  *
SLOWDOWN=12,       BUFFER THRESHOLD BELOW WHICH SLOWS  *
TRANSFR=18,      MAX BUFFERS PER PIU (SUPPORT 4K PIU)  *
TYPYSYS=MVS,       GENERATED UNDER AN MVS HOST          *
TYPGEN=NCP,        CHANNEL ATTACHED NCP                *
T1TIMER=(2.5,8.0), TOKEN-RING LOGICAL LINK REPLY TIMEOUT  *
T2TIMER=(0.5,1.5), TOKEN-RING LOGICAL LINK ACK TIMERS   *
USGTIER=4,         4 LSS, 1 HSS, 1TRA, 2 CA'S          *
VERSION=V7R3,      *
VRPOOL=(16,4),     *
VRTIMER0=(60,0,0), *
VRTIMER1=(60,0,0), *
VRTIMER2=(60,0,0)  *

*
*****
*      TOKEN RING DEFINITIONS
*****
G12TRP00 GROUP ECLTYPE=(PHYSICAL,PERIPHERAL), *
      TYPE=NCP *
      DIAL=NO, *
      LNCTL=SDLC, *
      MAXPU=1, *
      NPACOLL=(YES,EXTENDED), NPA COLLECTION OPTION *
      PUTYPE=1, *
      PUDR=NO, *
      LEVEL2=ECLNARL2, *
      LEVEL3=ECLNARL3, *
      LEVEL5=NCP, *
      TIMER=(ECLNART1,,ECLNART2,ECLNART3), *
      XIO=(ECLNARXL,ECLNARXS,ECLNARXI,ECLNARXK), *
      USERID=(5668854,ECLRBDT,NORECMS,,ECLNMVT), *
      SPEED=9600, *
      COMPTAD=YES, *
      COMPSWP=YES, *
      COMPOWN=YES *
L12TIC01 LINE ADDRESS=(1088,FULL), *
      LOCADD=400011600000, *
      MAXPU=1, *
      PORTADD=0, *
      MAXTSL=2042, *
      RCVBUFC=4095, *
      ADAPTER=TIC2, *
      TRSPEED=16, *
      UACB=(X$P1AX,X$P1AR) *
P12TIC01 PU ADDR=01, *
      INNPORT=YES, *
      ANS=CONT *

*

```

Note the following about this example:

- The MAXDATA parameter in the PCCU0112 definition sets the maximum data size for connecting to a 37x5 front-end-processor as an adjacent link station. The MAXDATA parameter on the host maps to the <max_btu_size> value set using the -APPN AdjLinkSta parameter on the NETBuilder bridge/router.
- The NETID parameter in the PCCU0112 definition is where you obtain the network ID required to connect to an APPN network.
- The TRANSFR parameter in the N12NCP definition maps to the SendWindow value set for both the -APPN SdlcAdjLinkSta and -APPN DlurLinkSta parameters. The window size only applies when the link station supports SDLC.
- The LOCADD parameter from the L12TIC01 line address is the MAC address used for the adjacent link station definition of a front-end-processor.

Example 5: Defining a Host as an SDLC Link Station

This example shows how to define an adjacent SDLC link station in the APPN service. This is a generic type SDLC node that does not have any dependent LUs that require DLUr. The host definition in this case is for a Type 2 PU (PU2). This is for an OS/2 workstation attached using SDLC (doing SDLC conversion), while defining the LUs as independent.

The following is the configuration on the NETBuilder bridge/router:

```
===== SHow -APPN SdlcAdjLinkSta =====
-----SDLC Adjacent Link Stations-----
No SDLC Adjacent Link Station Configured
===== -APPN SdlcAdjLinkSta parameter menu (Level 3)=====
  1 - Add
  2 - Delete
Select (1-2) ... <CR> to Exit =====> 1
Enter !<port> (mandatory) => 1
  Add      !<port> SdlcAdjLinkSta <type>(NN|EN|Learn)
<max_btu_size>(99-8912)
  <station addr>(Hex 1-FE) [CPName=<[netid.]cpname>] [Nodeid=<ID>]
[LinkName=<name>] [TGprof=<name>] [CPSess=(Yes|No)] [AutoStart=(Yes|No)]
[HPR=(Yes|No)] [SendWindow=<num>] [ContactTimer=<num>] [NoRspTimer=<num>]
[NoRsptimRetry=<num>]

Add !1 SdlcAdjLinkSta NN 2057 01 CPName=P10TRCP2 LinkName=G10TRL02
CPSess=Yes AutoStart=Yes HPR=Yes SendWindow=7
```

The following is the configuration required on the host (entries underlined in the host example map directly to the configuration required on the NETBuilder bridge/router):

```
*****
* THIS MEMBER CONTAINS VTAM SWITCHED NODE DEFINITIONS
* FOR TPNS TOKEN RING TESTING
*
* LIBRARY: NET.VTAMLST
* MEMBER: SWCPPU21
*
* CHANGE HISTORY:
* 08/16/95 (LDT): TEST TOKEN RING SCRIPT
*****
* 00010000
* 00020000
* 00030000
* 00040000
* 00050000
* 00060005
* 00070000
* 00080000
* 00090000
* 00091000
```



```

SWCPPU21 VBUILD TYPE=SWNET,MAXGRP=4,MAXNO=20          00092018
*                                                       00093000
P10TRPU1 PU ADDR=01,                                  X00094009
                CONNTYPE=LEN,                          X00094102
                CPNAME=P10TRCP1,                       X00094313
                DISCNT=NO,                              X00095000
                DYNLU=YES,                              X00095302
                IDBLK=999,                              X00096000
                IDNUM=01001,                            X00097000
                MAXDATA=265,                            X00098000
                MAXOUT=7,                               X00099000
                MAXPATH=1,                              X00100000
                NETID=US3COMHQ,                         X00110002
                PACING=7,                               X00120000
                PASSLIM=7,                              X00130000
                PUTYPE=2,                               X00140000
                SSCPFM=USSSCS,                          X00150000
                USSTAB=ISTINCDT,                        X00160000
                VPACING=7                               00170000
                PATH DIALNO=0104400037451088,GRPNM=G10TRL01 00180019
SLUDEI1 LU LOCADDR=01,DLOGMOD=SNX32702                00190019
SLUDEI2 LU LOCADDR=02,DLOGMOD=SNX32702                00191019
SLUDEFR LU LOCADDR=03,MODETAB=TPNSMTAB,DLOGMOD=FTPPS 00192021
SLUDEFS LULOCADDR=04,MODETAB=TPNSMTAB,DLOGMOD=FTPPS 00193021
*                                                       00200000
P10TRPU2 PUADDR=01,                                  X00210009
                CONNTYPE=APPN,                          X00220002
                CPCP=YES,                               X00220102
                CPNAME=P10TRCP2,                       X00220213
                DISCNT=NO,                              X00221002
                DYNLU=YES,                              X00223002
                IDBLK=999,                              X00230000
                IDNUM=01002,                            X00240000
                MAXDATA=2057,                           X00250016
                MAXOUT=7,                               X00260000
                MAXPATH=1,                              X00270000
                NETID=US3COMHQ,                         X00271002
                PACING=7,                               X00290000
                PASSLIM=7,                              X00300000
                PUTYPE=2,                               X00310000
                VPACING=7                               00340000
                PATH DIALNO=0104400037451089,GRPNM=G10TRL02 00341019
*                                                       00370000

```

Note the following about this example:

- The setting for the NETBuilder <max_btu_size> value must match that of the MAXDATA parameter in the PU definition (see the definition for P10TRPU2).
- The CPName value entered on the NETBuilder must match that of the CPNAME= PARAMETER in the PU definition (see the definition for P10TRPU2).

The CP name used is not fully-qualified, and as a result, the default NETID of the NETBuilder will be used.

- The LinkName entered on the NETBuilder in this case comes from the GRPNM parameter in the PU definition (see the definition for P10TRPU2).
- The SendWindow value entered on the NETBuilder is taken from the MAXOUT / PACING / PASSLIM parameters in the PU definition (see the definition for P10TRPU2).

Example 6: Mapping an SDLC DLUR Link Station to a Host SDLC PU Definition

This example shows how to map an SDLC DLUR link station in the APPN service to a host definition of an SDLC PU. The host definition in this case is for a Type 2 PU (PU2). This is for a workstation attached using SDLC (doing SDLC conversion).

The following is the configuration on the NETBuilder bridge/router:

```
===== SHow -APPN SdlcDlurLinkSta =====
-----SDLC Dlur Link Stations-----
No SDLC Dlur Link Station Configured
===== -APPN SdlcDlurLinkSta parameter menu (Level 3)=====
  1 - Add
  2 - Delete
Select (1-2) ... <CR> to Exit ====> 1
Enter !<port> (mandatory) => 1
  Add      !<port> SdlcDlurLinkSta <max_btu_size>(265-8912) <station
addr>(Hex 1-FE) <dspu name> [Nodeid=<ID>] [LinkName=<name>]
[Plus=<[netid.]name>] [Backup=<[netid.]name>] [TGprof=<name>]
[AutoStart=(Yes|No)] [PU2=(Yes|No)] [HPR=(Yes|No)] [SendWindow=<num>]
[ContactTimer=<num>] [NoRspTimer=<num>] [NoRspTimRetry=<num>]

ADD !1 SdlcDlurLinkSta 265 01 P10TRPU1 LinkName=G10TRL01
Plus=US3COMHQ.HOST3COM Backup=US3COMHQ.VTAM9370 AutoStart=No PU2=Yes
HPR=No SendWindow=7
```

The following is the configuration required on the host:

```
SWCPPU21 VBUILD TYPE=SWNET,MAXGRP=4,MAXNO=20          00092018
*                                                    00093000
P10TRPU1 PUADDR=01,                                  X00094009
                                                    X00094102
CONNTYPE=LEN,
CPNAME=P10TRCP1,                                     X00094313
DISCNT=NO,                                           X00095000
DYNLU=YES,                                           X00095302
IDBLK=999,                                           X00096000
IDNUM=01001,                                         X00097000
MAXDATA=265,                                         X00098000
MAXOUT=7,                                            X00099000
MAXPATH=1,                                           X00100000
NETID=US3COMHQ,                                     X00110002
PACING=7,                                            X00120000
PASSLIM=7,                                           X00130000
PUTYPE=2,                                            X00140000
SSCPFM=USSCS,                                       X00150000
```

```

USSTAB=ISTINCDT, X00160000
VPACING=7 00170000
PATH DIALNO=0104400037451088,GRPNM=G10TRL01 00180019
SLUDEI1 LU LOCADDR=01,DLOGMOD=SNX32702 00190019
SLUDEI2 LU LOCADDR=02,DLOGMOD=SNX32702 00191019
SLUDEFR LULOCADDR=03,MODETAB=TPNSMTAB,DLOGMOD=FTPSS 00192021
SLUDEFS LULOCADDR=04,MODETAB=TPNSMTAB,DLOGMOD=FTPSS 00193021
* 00200000

```

Note the following about this example:

- The dspu name entered on the NETBuilder bridge/router comes from the PUNAME from the PU definition.
- The dlus value entered on the NETBuilder bridge/router comes from the SSCPNAME=HOST3COM that is in the VTAM start options (ATCSTRxx).
- The Backup value entered on the NETBuilder bridge/router would come from the same parameter in the other backup VTAM.
- The HPR=No value entered on the NETBuilder bridge/router indicates that the PU is a type 2.0 node (non-HPR).

Example 7: Mapping a Default DLUs to the VTAM Start Options

This example shows how to map a Dependent LU server (VTAM) in the APPN service to the start options for that VTAM. By setting the default DLUs and configuring the corresponding VTAM start options, you will configure the defaults necessary for the VTAM host to start an APPN session with the NETBuilder bridge/router.

The following is the configuration on the NETBuilder bridge/router:

```

===== SHow -APPN DlurDefaults =====
-----DLUR Defaults-----
DLUS name = US3COMHQ.HOST3COM Backup name =
===== -APPN DlurDefaults parameter menu (Level 3)=====
SetD DlurDefaults = [Dlus=<[netid.]name|UNdef>]
[Backup=<[netid.]nam]
SetD DlurDefaults = Dlus=USCOMHQ.HOST3COM

```

The following is the configuration required on the host:

```

SSCPID=01,NOPROMPT, X00010000
CONFIG=01,MAXSUBA=63,SUPP=NOSUP, X00020002
HOSTSA=1, X00030007
SSCPNAME=HOST3COM, X00040001
NETID=US3COMHQ, X00050000
APPNCOS=NONE, X00050106
BN=YES,BNDYN=FULL, X00051003
CDSERVR=YES, X00052003
CONNTYPE=APPN, X00053003
CPCP=YES, X00054003
DYNADJCP=YES, X00055003

```

```

DYNPU=YES,                                X00055035
DYNLU=YES,                                X00055109
INITDB=NONE,                              X00056003
IOINT=600,                                X00056110
NCPBUFSZ=2048,                            X00056211
NODETYPE=NN,                              X00057003
SONLIM=( 40, 30 ),                        X00058018
SORDER=APPN,                              X00058118
TNSTAT,NOCNSL,TIME=15,                   X00059012
IOBUF=(1500,1016,18,,12,20),             X00060018
BSBUF=( 600,,14)                          00061017

```

Note the following about this example:

- The Dlus value entered on the NETBuilder bridge/router maps to the SSCPNAME=HOST3COM entry in the VTAM start options menu.
- The Backup value entered on the NETBuilder bridge/router comes from the same parameter entered on another VTAM uses as a backup.
- The DYNLU=YES entry in the VTAM start options mean that LUs do not have to be predefined with DLUR.
- The DYNADJCP=YES entry in the VTAM start options indicates support for dynamic adjacent CPs, meaning that new NETBuilder bridge/router network nodes can be added to the network without statically configuring them as adjacent link stations on VTAM.
- The SORDER=APPN entry in the VTAM start options indicates that the VTAM host will serve requests from APPN networks before other types of networks.

Example 8: Defining an LU Directory Entry

This example definition shows how to define an LU directory entry in the APPN service. Use this configuration to explicitly define an SNA resource location, to avoid the search process and only perform a locate.

The following is the configuration on the NETBuilder bridge/router:

```

===== SHow -APPN DirectoryEntry =====
-----Directory Entry-----
No Directory Entry Configured
===== -APPN DirectoryEntry parameter menu (Level 3)=====
  1 - Add
  2 - Delete
Select (1-2) ... <CR> to Exit =====> 1
  Add   DirectoryEntry <[netid.]resource name><type>
(LU|EN|NN|Wild)[[netid.]parent_name parent_type(EN|NN)] [[netid.]grandp]
  Add   DirectoryEntry US3COMHQ.LUFRED12 LU US3COMHQ.GORILLA EN

```

The following is the configuration required on the host:

```

000200 *           THIS MEMBER CONTAINS VTAM SWITCHED MAJOR NODE           *
000300 *           STATEMENTS FOR DLUR FOR JOHN SMITH                     *
000400 *                                                                 *
000500 *           CHANGE HISTORY:                                           *

```

```

000600 *          06/26/96 (JSS): DEFINED PUNAMES TO JOHNPU1          *
000706*****
000707          SWDLUR  VBUILD TYPE=SWNET,MAXGRP=2,MAXNO=2,MAXDLUR=10
000708 * 3174C APPN DLUR
000709  JOHNPU1  PU ADDR=01,          X
000710          ANS=CONT,          X
000711          DLOGMOD=SNX32702,  X
000720          DISCNT=NO,        X
000730          DYNLU=YES,        X
000740          IDBLK=017,        X
000750          IDNUM=9079D,      X
000760          IRETRY=YES,       X
000770          ISTATUS=ACTIVE,   X
000780          MAXDATA=521,      X
000790          MAXOUT=7,        X
000800          MAXPATH=2,        X
000900          PACING=0,        X
001000          PASSLIM=7,       X
001100          SSCPFM=USSSCS,    X
001200          USSTAB=USSTEST,  X
001300          VPACING=0
001400  JOHN1PT  PATH PID=1,      X
001500          DLURNAME=GORILLA,  X
001600          DLCADDR=(1,C,INTPU), X
001700          DLCADDR=(2,X,0179079D)
001800  LUJOHN12  LU LOCADDR=2
001900  LUJOHN13  LU LOCADDR=3
002000  LUJOHN14  LU LOCADDR=4
002100  LUJOHN15  LU LOCADDR=5
002101 *

```

Example 9: Mapping an SNA Class of Service (COS) to a Specific Transmission Priority

This is an example of mapping an SNA COS to a particular transmission priority in the APPN service. This allows you to obtain granularity in your path costs in an APPN network.

The following is the configuration on the NETBuilder bridge/router:

```

===== SHOW -APPN COSNodeRow =====
===== -APPN COSNodeRow parameter menu (Level 3)=====
  1 - Add
  2 - Delete
Select (1-2) ... <CR> to Exit ===== 1
  Add      COSNodeRow    <cos name> <weight>(0-255)
[Congestion=min(Yes|No),max(Yes|No)] [Resistance=min,max]
  Add      COSNodeRow    #INTER 30 C=(0,0) R=(0,31)

```

The following is the configuration required on the host (entries underlined in the host example map directly to the configuration required on the NETBuilder bridge/router):

```

173000  #INTER  APPNCOS PRIORITY=HIGH      transmission priority
174000  LINEROW WEIGHT=30,           line row weight          *
174500  NUMBER=1,                          line row number          *
175000  UPARAM1=(0,255),                     user defined char 1     *
175500  UPARAM2=(0,255),                     user defined char 2     *
176000  UPARAM3=(0,255),                     user defined char 3     *
176500  CAPACITY=(4M,MAXIMUM),               line speed              *
177000  COSTTIME=(0,0),                       cost per connect time   *
177500  COSTBYTE=(0,0),                       cost per byte transmitted *
178000  PDELAY=(MINIMUM,NEGLIGIB),propagation delay *
178500  SECURITY=(UNSECURE,MAXIMUM) security level for TG
179000  NODEROW NUMBER=1,                   node row number         *
179500  WEIGHT=5,                           node row weight         *
180000  CONGEST=(LOW,LOW),                   congestion              *
180500  ROUTERES=(0,31)                   route addition resistance
181000  LINEROW WEIGHT=60,           line row weight          *
181500  NUMBER=2,                          line row number         *
182000  UPARAM1=(0,255),                     user defined char 1     *
182000  UPARAM1=(0,255),                     user defined char 1     *
182500  UPARAM2=(0,255),                     user defined char 2     *
183000  UPARAM3=(0,255),                     user defined char 3     *
183500  CAPACITY=(56000,MAXIMUM),line speed *
184000  COSTTIME=(0,0),                       cost per connect time   *
184500  COSTBYTE=(0,0),                       cost per byte transmitted *
185000  PDELAY=(MINIMUM,TERRESTR),propagation delay *
185500  SECURITY=(UNSECURE,MAXIMUM) security level for TG
186000  NODEROW NUMBER=2,                   node row number         *
186500  WEIGHT=10,                          node row weight         *
187000  CONGEST=(LOW,LOW),                   congestion              *
187500  ROUTERES=(0,63)                   route addition resistance

```

Note the following about this example:

- This is not a complete ISTCOSxx (class of service) table.
- The Congestion value on the NETBuilder bridge/router maps to the CONGEST= parameter of the LINEROW statement.
- The Resistance value on the NETBuilder bridge/router maps to the ROUTERES= parameter of the node row statement.

Example 10: Mapping an SNA Class of Service to the APPN Service

This example shows how to map an SNA Class of Service definition to the APPN service.

The following is the configuration on the NETBuilder bridge/router:

```

===== SHow -APPN ConfigCOS =====
===== -APPN ConfigCOS parameter menu (Level 3)=====
      1 - Add
      2 - Delete
Select (1-2) ... <CR> to Exit =====> 1
      Add      ConfigCOS      <cos name> <transmit priority> [SNA defined
COS name]
      Add      ConfigCOS
    
```

The following is the configuration required on the host

```

*****
*
* MEMBER NAME: COSAPPN
*
* Descriptive name: IBM-Supplied APPN Class of Service Definitions
*
* STATUS: ACF/VTAM VERSION 4 RELEASE 2
*
* COPYRIGHT: LICENSED MATERIALS - PROPERTY OF IBM
*
* 5695-117 (C) COPYRIGHT IBM CORP. 1992.
* ALL RIGHTS RESERVED.
*
* U.S. GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
*
* SEE COPYRIGHT INSTRUCTIONS.
*****
*
*
*
#BATCH APPNCOSPRIORITY=LOW
          transmission priority          28800000
          LINEROW WEIGHT=30,             line row weight          *28900000
          NUMBER=1,                     line row number          *28950000
          UPARM1=(0,255),                user defined char 1      *29000000
          UPARM2=(0,255),                user defined char 2      *29050000
          UPARM3=(0,255),                user defined char 3      *29100000
          CAPACITY=(56000,MAXIMUM), line speed          *29150000
          COSTTIME=(0,0),                 cost per connect time    *29200000
          COSTBYTE=(0,0),                 cost per byte transmitted *29250000
          PDELAY=(MINIMUM,MAXIMUM), propagation delay      *29300000
          SECURITY=(UNSECURE,MAXIMUM) security level for TG      29350000
          NODEROW NUMBER=1,              node row number          *29400000
    
```

| | | |
|-----------------------------|---------------------------|-----------|
| WEIGHT=5, | node row weight | *29450000 |
| CONGEST=(LOW,LOW), | congestion | *29500000 |
| ROUTERES=(0,31) | route addition resistance | *29550000 |
| LINEROW WEIGHT=60, | line row weight | *29600000 |
| NUMBER=2, | line row number | *29650000 |
| UPARM1=(0,255), | user defined char 1 | *29700000 |
| UPARM2=(0,255), | user defined char 2 | *29750000 |
| UPARM3=(0,255), | user defined char 3 | *29800000 |
| CAPACITY=(19200,MAXIMUM), | line speed | *29850000 |
| COSTTIME=(0,0), | cost per connect time | *29900000 |
| COSTBYTE=(0,0), | cost per byte transmitted | *29950000 |
| PDELAY=(MINIMUM,MAXIMUM), | propagation delay | *30000000 |
| SECURITY=(UNSECURE,MAXIMUM) | security level for TG | 30050000 |
| NODEROW NUMBER=2, | node row number | *30100000 |
| WEIGHT=10, | node row weight | *30150000 |
| CONGEST=(LOW,LOW), | congestion | *30200000 |
| ROUTERES=(0,63) | route addition resistance | 30250000 |

Note the following about this example:

- This is not a complete ISTRCOSxx (class of service) table.
- The cos name value entered on the NETBuilder bridge/router refers to a local name that is mapped to an entry in ISTRCOSxx.
- The SNA defined COS name value entered on the NETBuilder bridge/router refers to the name of the COS entry (in this case the #BATCH statement from the ISTRCOSxx table in VTAMLST on the host).

BSC Host Example

This section provides a sample host configuration for configuring BSC between a host and a NETBuilder bridge/router.

The following is the configuration on the NETBuilder bridge/router:

```
===== SHow -BSC CONFIguration =====
-----BSC Configuration-----
!4 Control = Disable
!4 Role = Primary
Port CU_Name CU_Addr Local_Mac Remote_Mac Local_Sap Remote_Sap
4 P12021 1(0xC1) 400000003271 400011600000 0x4 0x4

!P12021 CUCONTrol = Disable
```

The configuration required on the host is shown below (entries underlined in the host example map directly to the configuration required on the NETBuilder bridge/router). Note the following about the BSC host configuration:

- You must specify that NCP will use the general polling procedure for this station and you must specify the polling characters to be assigned to the control unit of the station. If you omit GPOLL, devices must be polled individually.

- GPOLL is required if this CLUSTER definition statement represents an IBM 3271. For the ADDR keyword of each TERMINAL definition statement that defines a 2980, code the addressing characters assigned to that 2980. Because 2980s cannot be individually polled, the GPOLL keyword is not valid.

```

*****
* FROM 'SYS1.VTAMLST(N12V02)' NCP FOR BSC TESTING
* MVS RDO USED TO DO TESTING OF BSC TRANSPORT
*****
* BSC 3780 DEFINITIONS
*****
*
G12BSC1 GROUP      LNCTL=BSC,          BSC PROTOCOL          *
                  CLOCKNG=EXT,        EXTERNALLY CLOCKED MODEMS *
                  CODE=EBCDIC,         TRANSMISSION CODE      *
                  CU=2701,             EMULATE 2701 ** EP MODE ** *
                  DIAL=NO,            LEASED LINES          *
                  DIRECTN=INOUT,       NCP WIL SEND AND RECEIVE *
                  DLOGMOD=D4B32782,    NON-SNA 3270 24 X      *
                  DUPLEX=FULL,         RTS ACTIVE WHEN NCP RCV OR XMT *
                  ETRATIO=30,          3.0 % ERROR TO TRANSMISSION RATIO *
                  ISTATUS=ACTIVE,      ACTIVATE ALL RESOURCES  *
                  REPLYTO=3,           WAIT TIME FOR RESPONSE  *
                  USSTAB=USSTEST       UNFORMATTED SESSION SERVICES TABLE
*
*
*****
* PORT 00 BSC
*
*****
*
L1200 LINE ADDRESS=(000,42-0),      PORT ZERO ON UNIT=242      *
      USE=EP,                       INITIALLY OPERATING IN EP MODE *
      SPEED=9600,                    *
      NEWSYNC=NO
*
*
T1200140 TERMINAL TERM=3780
*
*****
* PORT 01 BSC
*
*****
*
L1201 LINE ADDRESS=(001,43-0),      PORT ZERO ON UNIT=243      *
      USE=EP,                       INITIALLY OPERATING IN EP MODE *
      SPEED=19200,                   *
      NEWSYNC=NO
*
*
T1201140 TERMINAL TERM=3780
*
*****
* BSC 3270 DEFINITIONS
*

```

```

*****
*
G12BSC2 GROUP      LNCTL=BSC,          BSC PROTOCOL          *
                   CLOCKNG=EXT,        EXTERNALLY CLOCKED MODEMS *
                   CODE=EBCDIC,         TRANSMISSION CODE      *
                   CU=2701,             EMULATE 2701    ** EP MODE ** *
                   CUTYPE=3271,        BSC CONTROLLER    ** EP MODE ** *
                   DIAL=NO,            LEASED LINES          *
                   DIRECTN=INOUT,      NCP WILL SEND AND RECEIVE *
                   DLOGMOD=D4B32782,  NON-SNA 3270 24 X 80   *
                   DUPLEX=FULL,       RTS ACTIVE WHEN NCP RCV OR XMT *
                   ETRATIO=30,        3.0 % ERROR TO TRANSMISSION RATIO *
                   ISTATUS=ACTIVE,    ACTIVATE ALL RESOURCES *
                   NPACOLL=YES,       NPA COLLECTION OPTION *
                   PAUSE=0,           SRVC ORDER TBL POLL CYCLE PAUSE *
                   POLIMIT=(10,QUEUE) ACCEPT 10 NACKS MAX *
                   POLLED=YES,        POLLED DEVICES        *
                   REPLYTO=3,         WAIT TIME FOR RESPONSE *
                   SERVLIM=50,        SERVICE ORDER TABLE NORMAL SCAN LIMIT *
                   USSTAB=USSTEST     UNFORMATTED SESSION SERVICES TABLE

*
*****
* PORT 02 BSC
*
*****
*
L1202      LINE ADDRESS=(002,44-0),    PORT ZERO ON UNIT=244 *
                   USE=NCP,           INITIALLY OPERATING IN NCP MODE *
                   SPEED=9600,        *
                   NEWSYNC=NO

*-----*
* LINE 02 (PORT 02) - CU 1:
* 4 TERMS SUPPORTING TYPE 2 COMPATIBLE SESSIONS
*-----*
P12021    CLUSTER CUTYPE=3271,        CONTROLLER RESPONDS TO THIS POLL *
                   GPOLL=40407F7F     CONTROLLER RESPONDS TO GENERAL POLL

T1202140  TERMINAL TERM=3277,ADDR=60604040,POLL=40404040
T12021C1  TERMINAL TERM=3277,ADDR=6060C1C1,POLL=4040C1C1
T12021C2  TERMINAL TERM=3277,ADDR=6060C2C2,POLL=4040C2C2
T12021C3  TERMINAL TERM=3277,ADDR=6060C3C3,POLL=4040C3C3
*

```

```

01680000
01690000
01880000
01890000
01900000
01910000
01920000
01930000
01980000
01990000
02000000
02010000
02080000

```

NetView Run Commands Support

You can configure the NETBuilder bridge/router to support NetView run commands using the parameters in the SNAMS Service. For more information, see the SNAMS Service Parameters chapter in *Reference for Enterprise OS Software*.

To issue run commands from NetView to access the NETBuilder bridge/router, enter RUNCMD followed by:

- The NETBuilder PU name as defined in VTAM and the NETBuilder SNA configuration
- The text "APPL=SNAMS"
- The individual NETBuilder bridge/router command being issued.

For example, to issue a run command from NetView to a PU named "PUSNAMS" and to enter the NETBuilder bridge/router command SHow -SYS VERsion, enter the following command on NetView:

```
RUNCMD SP=PUSNAMS, APPL=SNAMS, SHO -IP CONF
```

To issue a run command from NetView when the NETBuilder -SNAMS RunCmdSecurity parameter is enabled, you must enter the NETBuilder password. For example, to issue a run command from NetView to a PU named "PU01BJ1", to enter the password "MYPASSWORD", and to enter the NETBuilder bridge/router command SHow -IP CONFiguration, enter the following command on NetView:

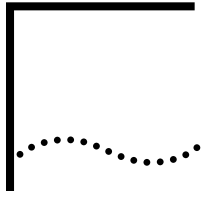
```
RUNCMD SP=PU01BJ1, APPL=SNAMS, PASSWD=MYPASSWORD, SHO -IP CONF
```



*To access the NETBuilder bridge/router, you must enter the text "APPL=SNAMS". Also, the password on the bridge/router must be configured using all upper-case letters if you plan to access the bridge/router from the host using NetView. If the bridge/router password is not configured in all upper-case letters, you will be unable to access the bridge/router interface from the host using NetView. For information about the PassWord command, see the Commands chapter in *Reference for Enterprise OS Software*.*



CAUTION: *When you configure the 327x host interface for NetView, the normal screen size for Model2 (MOD2) is 24 x 80 characters. To successfully view the NETBuilder bridge/router displays, change the 327x display mode to MOD5, which provides a screen display of 27 x 132 characters.*



ABBREVIATIONS AND ACRONYMS

This appendix provides a list of the abbreviations and acronyms used in this guide and corresponding NETBuilder documentation.

| | Abbreviation/Acronym | Meaning |
|----------|-----------------------------|--|
| A | AAL | ATM adaptation layer |
| | AARP | AppleTalk Address Resolution Protocol |
| | ABR | area border router |
| | AC | access control (Access Control when referring to service name) |
| | AEP | AppleTalk Echo Protocol |
| | AFP | AppleTalk Filing Protocol |
| | AFI | authority format identifier |
| | AMP | Adapter Management Protocol |
| | ANR | Automatic Network Routing |
| | ANSI | American National Standards Institute |
| | API | application program interface |
| | APPC | Advanced Program-to-Program Communication |
| | APPN | Advanced Peer-to-Peer Networking |
| | ARE | All Routes Explorer |
| | ARP | Address Resolution Protocol |
| | ARPANET | Advanced Research Projects Agency Network |
| | AS | autonomous system |
| | ASBR | Autonomous System Boundary Router |
| | ASN | autonomous system number |
| | ATG | address translation gateway |
| | ATM | Asynchronous Transfer Mode |
| | ATP | AppleTalk Transaction Protocol |
| B | BAN | Boundary Access Node |
| | BBS | bulletin board service |
| | BDR | backup designated router |
| | BGP | Border Gateway Protocol |
| | BMA | broadcast multi-access |
| | BNN | Boundary Network Node |
| | BOD | bandwidth-on-demand |
| | BPDU | Bridge Protocol Data Unit |
| | BRI | basic rate interface |
| | BSC | Binary Synchronous Communication |
| | BSD | Berkeley Software Distribution |

| Abbreviation/Acronym | Meaning |
|----------------------|--|
| BSI | British Standards Institute |
| BTU | basic transmission unit |
| BUS | Broadcast and Unknown Server |
| C | |
| CBPDU | Configuration Bridge Protocol Data Unit |
| CBT | Core-Based Trees |
| CC | configuration change |
| CCITT | Consultative Committee for International Telegraph and Telephone |
| CCS | <ol style="list-style-type: none"> 1 common channel signaling (ISDN) 2 compact configuration store |
| CD | <ol style="list-style-type: none"> 1 carrier detect (signal) 2 compact disc 3 collision detection |
| CERT | Computer Emergency Response Team |
| CHAP | Challenge Handshake Authentication Protocol |
| CIDR | Classless Interdomain Routing Protocol |
| CLNP | Connectionless Network Protocol |
| CLNS | Connectionless Network Service |
| CN | connection network |
| COS | class of service |
| CP | control point |
| CR | carriage return |
| CRC | cyclic redundancy check |
| CS | communications server |
| CSMA | carrier sense multiple access |
| CSMA/CD | carrier sense multiple access/collision detection |
| CSNP | Complete Sequence Number Protocol Data Unit |
| CSU | channel service unit |
| CTS | clear to send |
| CU | control unit |
| CUG | closed user group |
| D | |
| DCD | data carrier detected |
| DCE | <ol style="list-style-type: none"> 1 data communications equipment (EIA expansion) 2 data circuit-terminating equipment (CCITT) 3 Distributed Computing Environment (OSF) |
| DD | double-density |
| DDP | Datagram Delivery Protocol |
| DEB | Destination Explicit Blocking |
| DEF | Destination Explicit Forwarding |
| DHCP | Dynamic Host Configuration Protocol |
| DIB | Directory Information Base |
| DIS | Designated Intermediate System |
| DIT | Directory Information Tree |

| Abbreviation/Acronym | Meaning |
|----------------------|--|
| DLC | data link control |
| DLCI | data link connection identifier |
| DLSw | data link switching |
| DLT | data link test |
| DLUr | dependent LU requester |
| DLUs | dependent LU server |
| DN | 1 distinguished name
2 directory number |
| DNS | Domain Name Service |
| DOD | dial-on-demand (3Com) |
| DR | designated router |
| DSA | Directory System Agent |
| DSAP | destination service access point |
| DSP | domain specific part |
| DSPU | downstream physical unit |
| DSR | data set ready |
| DSU | digital service unit |
| DTE | data terminal equipment |
| DTR | data terminal ready |
| DUA | Directory User Agent |
| DVMRP | Distance Vector Multicast Routing Protocol |
| DXI | data exchange interface |
| E | |
| EBCDIC | Extended Binary Coded Decimal Interchange Code |
| ECM | enter command mode |
| ECS | Ether Connect System |
| ED | extra-density |
| ELAN | Emulated LAN |
| EN | end node (APPN) |
| ERP | Echo Reply Protocol |
| ERQ | echo request |
| ES | end system |
| ESH | end system hello |
| ES-IS | End System-to-Intermediate System |
| ETSI | European Telecommunications Standards Institute |
| F | |
| FAP | File Access Protocol |
| FDDI | Fiber Distributed Data Interface |
| FEP | front end processor |
| FIT | fully initializing terminal |
| FS | frame status |
| FSE | full status enquiry |
| FTAM | File Transfer Access and Management |
| FTP | File Transfer Protocol |

| | Abbreviation/Acronym | Meaning | |
|----------|-----------------------------|--|---------------------------------|
| G | GOSIP | Government Open Systems Interconnection Profile | |
| | GSA | Government Services Administration | |
| H | HD | high-density | |
| | HDLC | high-level data link control | |
| | HPR | High Performance Routing (APPN) | |
| | HSS | high-speed serial | |
| | HSSI | High-Speed Serial Interface | |
| I | IANA | Internet Assigned Numbers Authority | |
| | ICD | International Code Designator | |
| | ICMP | Internet Control Message Protocol | |
| | ICP | Internet Control Protocol | |
| | IDI | initial domain identifier | |
| | IDP | 1 initial domain part (OSI)
2 Internet Datagram Protocol | |
| | IEN | Internet Engineering Notes | |
| | IETF | Internet Engineering Task Force | |
| | IGMP | Internet Group Management Protocol | |
| | IGP | Interior Gateway Protocol | |
| | IIH | Intermediate System-to-Intermediate System hello packet | |
| | ISIS | Integrated Intermediate System-to-Intermediate System | |
| | ILMI | Interim Local Management Interface | |
| | IP | Internet Protocol | |
| | IPC | interprocessor communication | |
| | IPX | Internetwork Packet Exchange | |
| | IS | intermediate system | |
| | ISDN | Integrated Services Digital Network | |
| | ISH | intermediate system hello | |
| | IS-IS | Intermediate System-to-Intermediate System | |
| | ISO | International Organization for Standardization | |
| | ISR | Intermediate Session Routing (APPN) | |
| | ITCM | Integrated T1 Controller Module | |
| | ITU-TSS | International Telecommunications Union–Telecommunications Standards Sector | |
| | L | LAA | LAN Address Administration |
| | | LAP | Link Access Procedure |
| | | LAPB | Link Access Procedure, Balanced |
| LANE | | LAN Emulation Client | |
| LAT | | local area transport | |
| LCN | | logical channel number | |
| LCP | | Link Control Protocol | |
| LEM | | Link Error Monitor | |
| LEN | | low-entry networking (APPN) | |
| LEC | | LAN Emulation Client | |

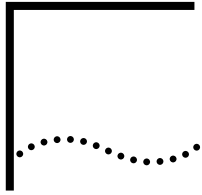
| Abbreviation/Acronym | Meaning |
|----------------------|---|
| LECS | LAN Emulation Configuration Server |
| LES | LAN Emulation Server |
| LF | linefeed |
| LFS | largest frame size |
| LIS | logical IP subnetwork |
| LLC | Logical Link Control |
| LLC2 | Logical Link Control, type 2 |
| LMF | Line Management Function |
| LMI | Local Management Interface |
| LNМ | LAN Net Manager |
| LS | link state |
| LSA | link state advertisement |
| LSP | <ol style="list-style-type: none"> 1 Link State Protocol 2 link state packets |
| LSR | link state information request |
| LSU | link state update |
| LU | logical unit |
| LUNI | LAN Emulation User Network Interface |
| M MAC | <ol style="list-style-type: none"> 1 media access control 2 media access controller (FDDI) |
| MAU | <ol style="list-style-type: none"> 1 multistation access unit (token ring) 2 medium access unit (Ethernet) |
| MIB | management information base |
| MIC | media interface connector |
| MIP | Multicast Internet Protocol |
| MLN | multiple logical networks |
| MLP | Multilink Protocol |
| MOSPF | Multicast Open Shortest Path First |
| MP | multiprocessor |
| MPATM | multiprotocol ATM |
| MSB | most significant bit |
| MTU | maximum transmission unit |
| N NA | Neighbor Acquisition |
| NBMA | non-broadcast multi-access interfaces |
| NBP | <ol style="list-style-type: none"> 1 Name Binding Protocol (AppleTalk) 2 NetBIOS Protocol (3Com) |
| NCE | network connection endpoint |
| NCP | <ol style="list-style-type: none"> 1 Network Control Protocol 2 NetWare Core Protocol (Novell) 3 Network Control Program (SNA) |
| NCS/AT | Network Control Server/AT |
| NET | Network Entity Title |

| Abbreviation/Acronym | Meaning |
|-----------------------------|--|
| NetBIOS | Network Basic Input/Output System |
| NFS | Network File System |
| NLPID | Network Layer Protocol Identifier |
| NLSP | NetWare Link Services Protocol |
| NMI | nonmaskable interrupt |
| NMS | Network Management System |
| NMU | Network Management Utilities |
| NN | network node (APPN) |
| NPDU | network protocol data unit |
| NR | neighbor reachability |
| NRIP | NetWare Routing Information Protocol (Novell) |
| NRZ | non-return to zero |
| NRZI | non-return to zero inverted |
| NSA | National Security Agency |
| NSAP | network service access point |
| NSF | 1 National Specific Facilities
2 National Science Foundation |
| NT1 | network termination 1 |
| O OSI | Open System Interconnection |
| OSIAPPL | Open System Interconnection Applications |
| OSPF | Open Shortest Path First |
| P PAD | packet assembler/disassembler |
| PAP | Password Authentication Protocol |
| PCM | 1 physical connection management
2 pulse code modulation (ISDN) |
| PDN | public data network |
| PDU | protocol data unit |
| PEP | partitioned emulation programming |
| PLU | primary logical unit |
| PMF | parameter management frame |
| PPM | port and path module |
| PPP | Point-to-Point Protocol |
| PPTP | Point-to-Point Tunneling Protocol |
| PRI | primary rate interface |
| PSAP | presentation service access point |
| PSDN | packet switching data network |
| PSNP | Partial Sequence Number PDU |
| PU | physical unit |
| PVC | permanent virtual circuit |
| Q QOS | quality of service |
| R RARP | Reverse Address Resolution Protocol |
| RD | 1 route designator |

| Abbreviation/Acronym | Meaning |
|----------------------|--|
| | 2 received data (signal) |
| RDP | Router Discovery Protocol |
| RFC | Request for Comments |
| RH | request/response header |
| RI | routing information |
| RIB | routing information database |
| RIF | routing information field |
| RII | routing information indicator |
| RIP | Routing Information Protocol |
| RIPIP | Routing Information Protocol for IP |
| RIPXNS | Routing Information Protocol for XNS |
| RLSD | received line signal detector |
| RMA | Return Materials Authorization |
| RMON | Remote Monitoring |
| RPB | Reverse Path Broadcasting |
| RPF | Reverse Path Forwarding |
| RPM | Reverse Path Multicasting |
| RSCV | Route Selection Control Vector |
| RTMP | Routing Table Maintenance Protocol |
| RTP | 1 Routing Table Protocol
2 routing update packets
3 Rapid Transport Protocol (APPN HPR) |
| RTS | request to send |
| RU | request/response unit |
| S SAP | 1 Service Advertising Protocol (NetWare)
2 service access point (OSI and SNA) |
| SAS | single-attached station |
| SDC | synchronous data compression |
| SDLC | Synchronous Data Link Control |
| SEB | Source Explicit Blocking |
| SEF | Source Explicit Forwarding |
| SIO | serial input/output |
| SIP | SMDS Interface Protocol |
| SLU | secondary logical unit |
| SMDS | Switched Multimegabit Data Service |
| SMT | Station Management |
| SMTP | Simple Mail Transfer Protocol |
| SNA | Systems Network Architecture |
| SNAP | Subnetwork Access Protocol |
| SNI | 1 Subscriber Network Interface
2 System Network Interconnection |
| SNMP | Simple Network Management Protocol |

| Abbreviation/Acronym | Meaning |
|-----------------------------|---|
| SNPA | Subnetwork Point of Attachment |
| SPF | shortest path first |
| SPID | Service Profile Identifiers |
| SPT | shortest path tree |
| SPX | sequenced packet exchange |
| SQL | Structured Query Language |
| SR | source route (Source Route when referring to the service) |
| SRF | specifically routed frame |
| SRT | source-route transparent |
| SRTG | source route transparent bridging gateway |
| SSAP | source service access point |
| SSCP | session services control point |
| STA | Spanning Tree Algorithm |
| STE | Spanning Tree Explorer |
| STP | Spanning Tree Protocol |
| SVC | switched virtual circuit |
| T | |
| TA | terminal adapter |
| TACACS | Terminal Access Controller Access Control System |
| TCAPPL | Transmission Control Protocol Applications |
| TCP | Transmission Control Protocol |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TERM | Terminal (service name) |
| TFTP | Trivial File Transfer Protocol |
| TG | transmission group |
| TH | transmission header |
| TOS | type of service |
| TRPB | Truncated Reverse Path Broadcasting |
| TTL | time-to-live |
| TUBA | TCP and UDP with Bigger Addresses |
| U | |
| UDP | User Datagram Protocol |
| UME | User-to-Network Interface Management Entity |
| UNI | user-to-network interface |
| V | |
| VC | virtual circuit (X.25) |
| | virtual connection (Frame Relay) |
| | virtual channel (ATM) |
| VCC | virtual channel connection |
| VCI | virtual channel identifier (APPN) |
| VCID | virtual circuit identifier |
| VIP | VINES Internet Protocol |
| VPI | virtual path identifier |
| VPN | virtual private network |
| VRN | virtual routing node |

| Abbreviation/Acronym | | Meaning |
|-----------------------------|-----|---------------------------|
| | VTP | Virtual Terminal Protocol |
| X | XID | exchange identification |
| | XNS | Xerox Network Systems |
| Z | ZIP | Zone Information Protocol |



TECHNICAL SUPPORT

3Com provides easy access to technical support information through a variety of services. This appendix describes these services.

Information contained in this appendix is correct at time of publication. For the most recent information, 3Com recommends that you access the 3Com Corporation World Wide Web site.

Online Technical Services

3Com offers worldwide product support 24 hours a day, 7 days a week, through the following online systems:

- World Wide Web site
- 3Com Knowledgebase Web Services
- 3Com FTP site
- 3Com Bulletin Board Service (3Com BBS)
- 3Com FactsSM Automated Fax Service

World Wide Web Site

To access the latest networking information on the 3Com Corporation World Wide Web site enter this URL into your Internet browser:

`http://www.3com.com/`

This service provides access to online support information such as technical documentation and software library, as well as support options that range from technical education to maintenance and professional services.

3Com Knowledgebase Web Services

This interactive tool contains technical product information compiled by 3Com expert technical engineers around the globe. Located on the World Wide Web at **`http://knowledgebase.3com.com`**, this service gives all 3Com customers and partners complementary, round-the-clock access to technical information on most 3Com products.

3Com FTP Site

Download drivers, patches, software, and MIBs across the Internet from the 3Com public FTP site. This service is available 24 hours a day, 7 days a week.

To connect to the 3Com FTP site, enter the following information into your FTP client:

- Hostname: **`ftp.3com.com`**
- Username: **`anonymous`**
- Password: **`<your Internet e-mail address>`**



You do not need a user name and password with Web browser software such as Netscape Navigator and Internet Explorer.

3Com Bulletin Board Service

The 3Com BBS contains patches, software, and drivers for 3Com products. This service is available through analog modem or digital modem (ISDN) 24 hours a day, 7 days a week.

Access by Analog Modem

To reach the service by modem, set your modem to 8 data bits, no parity, and 1 stop bit. Call the telephone number nearest you:

| Country | Data Rate | Telephone Number | Country | Data Rate | Telephone Number |
|-----------|------------------|------------------|----------------|------------------|------------------|
| Australia | Up to 14,400 bps | 61 2 9955 2073 | Japan | Up to 14,400 bps | 81 3 5977 7977 |
| Brazil | Up to 28,800 bps | 55 11 5181 9666 | Mexico | Up to 28,800 bps | 52 5 520 7835 |
| France | Up to 14,400 bps | 33 1 6986 6954 | P.R. of China | Up to 14,400 bps | 86 10 684 92351 |
| Germany | Up to 28,800 bps | 4989 62732 188 | Taiwan, R.O.C. | Up to 14,400 bps | 886 2 377 5840 |
| Hong Kong | Up to 14,400 bps | 852 2537 5601 | U.K. | Up to 28,800 bps | 44 1442 438278 |
| Italy | Up to 14,400 bps | 39 2 27300680 | U.S.A. | Up to 53,333 bps | 1 847 262 6000 |

Access by Digital Modem

ISDN users can dial in to the 3Com BBS using a digital modem for fast access up to 64 Kbps. To access the 3Com BBS using ISDN, call the following number:

1 847 262 6000

3Com Facts Automated Fax Service

The 3Com Facts automated fax service provides technical articles, diagrams, and troubleshooting instructions on 3Com products 24 hours a day, 7 days a week.

Call 3Com Facts using your Touch-Tone telephone:

1 408 727 7021

Support from Your Network Supplier

If you require additional assistance, contact your network supplier. Many suppliers are authorized 3Com service partners who are qualified to provide a variety of services, including network planning, installation, hardware maintenance, application training, and support services.

When you contact your network supplier for assistance, have the following information ready:

- Product model name, part number, and serial number
- A list of system hardware and software, including revision levels
- Diagnostic error messages
- Details about recent configuration changes, if applicable

If you are unable to contact your network supplier, see the following section on how to contact 3Com.

Support from 3Com

If you are unable to obtain assistance from the 3Com online technical resources or from your network supplier, 3Com offers technical telephone support services. To find out more about your support options, please the 3Com technical telephone support phone number at the location nearest you.

When you contact 3Com for assistance, have the following information ready:

- Product model name, part number, and serial number
- A list of system hardware and software, including revision levels
- Diagnostic error messages
- Details about recent configuration changes, if applicable

Here is a list of worldwide technical telephone support numbers:

| Country | Telephone Number | Country | Telephone Number |
|--|---|----------------------------|------------------------------------|
| Asia Pacific Rim | | | |
| Australia | 1 800 678 515 | P.R. of China | 10800 61 00137 or
021 6350 1590 |
| Hong Kong | 800 933 486 | Singapore | 800 6161 463 |
| India | +61 2 9937 5085 | S. Korea | |
| Indonesia | 001 800 61 009 | From anywhere in S. Korea: | 00798 611 2230 |
| Japan | 0031 61 6439 | From Seoul: | (0)2 3455 6455 |
| Malaysia | 1800 801 777 | Taiwan, R.O.C. | 0080 611 261 |
| New Zealand | 0800 446 398 | Thailand | 001 800 611 2000 |
| Pakistan | +61 2 9937 5085 | | |
| Philippines | 1235 61 266 2602 | | |
| Europe | | | |
| From anywhere in Europe, call: | +31 (0)30 6029900 phone | | |
| | +31 (0)30 6029999 fax | | |
| Europe, South Africa, and Middle East | | | |
| From the following countries, you may use the toll-free numbers: | | | |
| Austria | 0800 297468 | Netherlands | 0800 0227788 |
| Belgium | 0800 71429 | Norway | 800 11376 |
| Denmark | 800 17309 | Poland | 00800 3111206 |
| Finland | 0800 113153 | Portugal | 0800 831416 |
| France | 0800 917959 | South Africa | 0800 995014 |
| Germany | 0800 1821502 | Spain | 900 983125 |
| Hungary | 00800 12813 | Sweden | 020 795482 |
| Ireland | 1800 553117 | Switzerland | 0800 55 3072 |
| Israel | 1800 9453794 | U.K. | 0800 966197 |
| Italy | 1678 79489 | | |
| Latin America | | | |
| Argentina | AT&T +800 666 5065 | Mexico | 01 800 CARE (01 800 2273) |
| Brazil | 0800 13 3266 | Peru | AT&T +800 666 5065 |
| Chile | 1230 020 0645 | Puerto Rico | 800 666 5065 |
| Colombia | 98012 2127 | Venezuela | AT&T +800 666 5065 |
| North America | | | |
| | 1 800 NET 3Com
(1 800 638 3266) | | |
| | Enterprise Customers:
1 800 876-3266 | | |

Returning Products for Repair

Before you send a product directly to 3Com for repair, you must first obtain an authorization number. Products sent to 3Com without authorization numbers will be returned to the sender unopened, at the sender's expense.

To obtain an authorization number, call or fax:

| Country | Telephone Number | Fax Number |
|--|---|-----------------------------------|
| Asia, Pacific Rim | +65 543 6500 | +65 543 6348 |
| Europe, South Africa, and Middle East | +31 30 6029900 | +31 30 6029999 |
| Latin America | 1 408 326 2927 | 1 408 326 3355 |
| From the following countries, you may call the toll-free numbers: select option 2 and then option 2: | | |
| Austria | 0800 297468 | |
| Belgium | 0800 71429 | |
| Denmark | 800 17309 | |
| Finland | 0800 113153 | |
| France | 0800 917959 | |
| Germany | 0800 1821502 | |
| Hungary | 00800 12813 | |
| Ireland | 1800 553117 | |
| Israel | 1800 9453794 | |
| Italy | 1678 79489 | |
| Netherlands | 0800 0227788 | |
| Norway | 800 11376 | |
| Poland | 00800 3111206 | |
| Portugal | 0800 831416 | |
| South Africa | 0800 995014 | |
| Spain | 900 983125 | |
| Sweden | 020 795482 | |
| Switzerland | 0800 55 3072 | |
| U.K. | 0800 966197 | |
| U.S.A. and Canada | 1 800 NET 3Com
(1 800 638 3266) | 1 408 326 7120
(not toll-free) |
| | Enterprise Customers:
1 800 876 3266 | |

23 MAR 99

INDEX

Numbers

- 3Com bulletin board service (3Com BBS) 1496
- 3Com Knowledgebase Web Services 1495
- 3Com URL 1495
- 3ComFacts 1496

A

- AAL 1210
- abbreviations and acronyms 1485
- AccessAct parameter 743, 744
- AccessList parameter
 - L2Tunnel Service 290
- accumulated system statistics 1359
- adaptation layer
 - AAL3/4 1213, 1215
 - AAL5 1175, 1213, 1215
- Add Name menu 1148
- address
 - mapping, NAT Service 319
 - migration 330
 - redirection 331
- address mapping
 - AMP functional to multicast 1302
 - ATM DXI to Frame Relay DLCIs 1214
 - IPX to Frame Relay DLCIs 1215
 - SDLC devices 721
- ADDRess parameter 621
- Address Resolution Protocol. *See* ARP
- addressing
 - CU 717
 - SNA and VTAM setup 715
- AddrLOCation parameter 808
- adjacent link stations
 - activating and deactivating 517
 - configuration
 - defining characteristics 489, 500
 - defining for a port 487, 499
 - defining for an SDLC port 488
 - deleting 516
 - displaying
 - current status 521
 - list of 520
 - parallel TGs 505
- adjacent nodes
 - defining for LEN end nodes 503
 - defining network nodes and end nodes as 503
 - displaying current status 521
- AdjLenDef parameter 503
- AdjLinkSta parameter 487, 499, 520, 537
- AdjNodeStatus parameter 521
- Advanced Peer-to-Peer Networking. *See* APPN routing
- AdvertisePolicy parameter
 - DECnet 641
 - NRIP 605
 - SAP 607
- AdvToNeighbor parameter
 - DECnet 642
 - NRIP 584, 607
 - SAP 584, 607
- aggregation
 - BGP routes 188, 225
 - DVMRP routes 348
 - RIP routes 166
- All Routes Explorer frame. *See* ARE
- AllRoutes parameter
 - IDP 701
 - IPX 597
 - SR 156
- AllServers parameter 598, 601
- American National Standards Institute (ANSI) 1342
- AMP. *See* network management, Adapter Management Protocol (AMP) 1301
- AppleTalk Name Binding service, mapping to UDP ports 388
- AppleTalk routing
 - ADDRess parameter 621
 - broadcast packets, changing transmission interval 621
 - configuration
 - checking 617
 - prerequisites for 613
 - troubleshooting 618
 - CONFIguration parameter 620
 - configuring over
 - Frame Relay 1014 to 1016
 - LANs 614
 - non-AppleTalk data link 620
 - SMDS 1052 to 1056
 - X.25 1082 to 1085
 - CONTRol parameter 633
 - DefaultZone parameter 615, 617
 - description 628
 - entity filtering 625
 - entity names 630
 - EntityFilter parameter 626
 - EntityFilterNum parameter 626
 - filtering on Frame Relay ports 635
 - Macintosh extended character set 631
 - multiple seed routers, setting up 619
 - NetFilter parameter 623
 - NetFilterType parameter 623
 - NetRange parameter 615, 616
 - network number-based filtering 623 to 624
 - network operations 634
 - network-to-zone mapping, displaying 630
 - network topology example 635
 - network zones 629
 - NetZoneMapping parameter 617, 630
 - nonseed router 615 to 616
 - port startup 633
 - RouteAgingTime parameter 621
 - routes
 - learning 621
 - validity check interval, changing 621
 - RouteUpdateTime parameter 621
 - Routing Table Maintenance Protocol (RTMP) 636
 - routing table, displaying 635
 - SampleTime parameter 618
 - seed router
 - description 616, 629
 - setting up 615
 - split horizon 635
 - StartupNET parameter 633
 - StartupNODE parameter 633
 - statistics display 1359
 - STATistics parameter 618
 - WAN configurations 615
 - ZONE parameter 615, 616
 - ZoneNetMapping parameter 617, 630
- AppleTalk Service statistics 1359
- AppleTalk translation bridging restrictions 116
- AppleTalk zone information service, mapping to UDP ports 388
- APPN class of service
 - deleting class of service name 551
 - displaying 551
 - IBM standard defaults 549
 - mapping mode names to class of service names 551
 - transmission group rows 551
- APPN routing
 - activating and deactivating adjacent link stations 517
 - activating and deactivating ports 516
 - adjacent link stations 487, 499
 - AdjLenDef parameter 503
 - AdjLinkSta parameter 487, 499, 520, 537
 - AdjNodeStatus parameter 521
 - APPN ports
 - defining 484
 - defining for HPR 536
 - basic transmission unit (BTU), setting the maximum 532
 - ConfigCOS parameter 550
 - CONFIguration parameter 498
 - configuring
 - basic 481
 - Boundary Routing 510
 - bridge/router as network node 481
 - DLSw between nodes 508
 - Frame Relay for APPN 1016
 - local node name 484
 - parallel TGs 505
 - virtual ports for APPN over Frame Relay 1018
 - connection networks
 - boundary routing 513
 - configuration 513
 - using to scale large networks 511
 - CONNecion parameter 520
 - ConnNetworkChar parameter 513
 - ConnNetworkDef parameter 513
 - CONTRol parameter 493
 - COSDef parameter 551
 - COSNodeRow parameter 550
 - COSTgRow parameter 551
 - customizing 498
 - deleting
 - class of service name and node row 551
 - LEN end node directory entries 503

- links to adjacent nodes 516
- network node directory
 - entries 504
- transmission group row 552
- dependent LU support 490
 - configuring upstream links to DLUs 492
 - defining downstream links to PU 2.x nodes 492
 - defining the DLUs 491
- DirRectory parameter 518
- DirectoryEntry parameter 503
- disabling and reenabling the network node 515
- disabling the network node 515
- displaying
 - active connections 520
 - adjacent link station list 520
 - current adjacent link station status 521
 - current adjacent node status 521
 - current status of APPN ports 520
 - directory information 518
 - DLUr link stations 497
 - downstream LUs and PUs 497
 - Intermediate Session Routing status 521
 - network topology information 518
 - RTP information 540
 - transmission group information 519
 - upstream DLUs status 497
- displaying class of service 551
- DlurDefaults parameter 491
- DlurLinkSta parameter 492
- dynamic configuration options 494
- enabling the network node 493
- end node definition 525
- High Performance Routing
 - comparing to ISR 547
 - defining link stations for 537
 - defining ports for 536
 - designing HPR subnets 538
 - HPR timers 540
 - initiating non-disruptive path switch 540
 - non-disruptive path switch 544
 - RTP connection statistics 540
 - RTP display 540
- HprTimer parameter 540
- IBM class of service defaults 549
- IBM references 534
- ISRsessions parameter 497, 522
- LEN end node definition 525
- links to end nodes 499
- LinkStaCHar parameter 489, 500
- LinkStaCONTRol parameter 494, 517, 521
- LocalNodeName parameter 484
- LocalNodeResist parameter 484
- ModetoCosMap parameter 551
- network node
 - defined 524
 - defining adjacent link station characteristics 489, 500
 - defining adjacent link station to a port 487, 499
 - defining as adjacent node 503
 - directory 503
 - directory entries 500
 - operating 515
 - network node port 507
 - network node port definition 484
 - NNtopology parameter 519
 - node row for class of service, adding 550
 - node types 523
 - PathSwitch command 540
 - pinging to APPN resources 518
 - PortCONTRol parameter 516
 - PortDef parameter 536
 - RTP parameter 540
 - RTPStats parameter 540
 - SDLC DLUr link station configuration 492
 - SdclAdjLinkSta parameter 488, 499, 520, 537
 - TG parameter 519
 - TG row configuration 551
 - troubleshooting 497
- APpnPING command 518
- ARE 154
- Area Border Router (ABR) 208, 209
- ARP 225 to 226
- ARP Service statistics 1363
- asynch communications
 - baud rates supported for 806
 - configuration examples 811
 - configuring 805
 - defining CUs 809
- asynchronous communications. *See* asynch communications
- Asynchronous Transfer Mode 1210
- Asynchronous Transfer Mode (ATM) 1007
- Asynchronous Transfer Mode Data Exchange Interface. *See* ATM DXI
- Asynchronous Transfer Mode with LAN Emulation. *See* ATMLE Service
- Asynchronous Transfer Mode. *See* ATM
- ATM
 - adaptation layer, AAL5 1175
 - addressing 1187
 - cell size 1186
 - configuration checking 1178
 - configuring
 - ATM Service 1176
 - IP routing 1181
 - IPX routing 1184
 - traffic shaping attributes 1177, 1189
 - transparent bridging 1179
 - VCID for PVCs 1177
 - virtual port 1177
 - connection-oriented mode 1186
 - encapsulation types 1188
 - ILMI Protocol 1192
 - loopback testing 1178
 - monitoring the network 1178
 - network
 - interfaces 1186
 - management 1191
 - monitoring 1178
 - next-hop split horizon 1195
 - PermVirCircuit parameter 1177, 1188
 - prerequisites 1175
 - PVCs 1188
 - quality of service 1188
 - terminology 1195
- topologies
 - fully meshed 1192
 - nonmeshed 1193
 - partially meshed 1194
- TrafficShaper parameter 1189
- transparent bridging over 94
- UME 1191
- UNI 1186
- virtual
 - channel 1187
 - channel identifier (VCID) 1177
 - path 1187
 - ports, lack of connectivity 1195
 - VPI.VCI 1187
- ATM Adaptation Layer. *See* AAL
- ATM DXI
 - adaptation layer 1213, 1215
 - addresses 1214
 - configuring
 - IPX routing 1215
 - transparent bridging 1215
 - XNS routing 1215
 - DSU 1213
 - encapsulation type 1215, 1216
 - LMI Protocol 1215
 - UNI 1213
- ATMLE Service
 - addressing 1204
 - cell size 1203
 - configuring
 - ATMLE Service 709, 710, 711, 1197, 1198
 - virtual port 1198, 1199, 1201, 1202
 - connection-oriented mode 1203
 - network, interfaces 1204
 - prerequisites 709, 1197
 - setting up 709, 1197
 - terminology 1210
 - verifying the configuration 1200
 - VPI 1204
- AtmToFr utility 1214
- ATUN statistics display 1365
- audit trail messages
 - list 1413
- audit trail notification 1295
- AuditLog Service
 - function of 1289
 - sample messages display 1290
- authentication, PPP 908
- auto startup
 - attributes detected 888
 - BOOTP server 882, 884, 890
 - Boundary Routing 817, 822, 827
 - concepts 888 to ??
 - configuration files 886, 890
 - configuring 883, 884
 - description 888
 - network resiliency 841, 866
 - prerequisites 882
 - sample topologies 883
 - software tools 882
 - TFTP server 882, 886, 890
 - UDP Broadcast Helper 890
- automatic connections, incoming. *See* incoming connections
- automatic connections, outgoing. *See* outgoing connections
- Autonomous System Boundary Router (ASBR) 208, 210

B

- backup link, configuring 1031
- BackUp parameter
 - VRRP Service 368
- backup PVC, configuring 1030
- BACKwards command 1172
- bandwidth
 - dynamic, description 1004
 - static, description 1004
- bandwidth management 1004
 - See also* dial-up lines
 - command summary 1001
 - definition 1005
 - description 1004
 - manual mode
 - configuring 992
 - definition of 971
 - path configuration summary 970
 - status of 972
 - system mode
 - configuring 984
 - definition of 968
 - terms 1005
 - troubleshooting configuration 994
 - verifying configuration 993
- bandwidth-on-demand
 - See also* dial-up lines
 - configuring 984
 - description 969
- BGP
 - advantages 216
 - default route 187
 - DefaultMetric parameter 191
 - DefaultNet parameter 187
 - ExteriorPolicy parameter 192
 - importing routes from BGP
 - multihomed AS 191
 - stub AS 191
 - transit AS 192
 - importing routes from IGP 189
 - learning routes 216
 - path attributes
 - AGGREGATION 222
 - AS-PATH 219
 - ATOMIC-AGGREGATE 221
 - classifications 218
 - LOCAL-PREF 221
 - MULTI-EXIT-DISC 220
 - NEXT-HOP 220
 - ORIGIN 220
 - path selection 222
 - peers 216
 - peer-to-peer communication
 - phases 217
 - policies
 - AS-path permit or deny 193, 224
 - AS-path weight 195
 - degree of preference
 - examples 196
 - deny filter examples 194
 - exterior 224
 - interior 224
 - network number 192, 224
 - permit filter examples 194
 - weight filter examples 196
 - regular expressions 194, 1431
 - route aggregation 188, 225
 - BGP Service statistics 1366
- Binary Synchronous Communications (BISYNC). *See* BSC
- BISYNC. *See* BSC
- BLimitTimer parameter 141
- BODIncrLimit parameter 969
- BODThreshold parameter 969
- booting clients in order 394
- booting from specific servers 393
 - configuring server addresses 392
 - enabling 391
- BOOTP server 882, 884, 890
- BootpMaxHops parameter 393
- BootpThreshold parameter 393
- Bootstrap Protocol service, mapping to UDP ports 388
- Border Gateway Protocol. *See* BGP
- Boundary Routing
 - advantages 856 to 863
 - auto startup 881
 - central node
 - NETBuilder II 844, 847, 850, 852
 - SuperStack II 227 844, 848
 - SuperStack II 327 844, 854
 - SuperStack II 427 844, 848
 - SuperStack II 527 844, 854
 - configuring
 - dual PVCs for SNA traffic 865, 1028
 - for APPN 510
 - for Frame Relay 822
 - for PPP 817
 - for X.25 828
 - network resiliency 840
 - data compression 859
 - description 844
 - dial-up backup line 836, 862
 - environment, typical 847, 848
 - hardware, nodes 844, 845
 - IBM
 - advantages 856
 - APPN environment 855
 - configuring 817, 828
 - for Frame Relay 822**
 - data compression 859
 - environment, typical 850, 852, 854
 - exchanging data between peers 859
 - local termination 860
 - network resiliency 840, 865
 - prioritization, automatic 862
 - SDLC 856
 - smart filtering 856
 - troubleshooting 835
 - verifying configuration 833
 - IPX spoofing over dial-on-demand lines 859
 - legal topologies 845
 - network resiliency 840, 865
 - peripheral node 845
 - protocol islands 857
 - redundant links and routes 840, 865
 - smart filtering 856
 - topology, assigning network numbers to ports 864
 - troubleshooting 834
 - verifying configuration 832
- Boundary Routing of IBM Traffic Using SmartSwitching (BRITSS)
 - specification 836, 865
 - enabling on Frame Relay SNA PVC 838
- BoundaryAddr parameter 344
- bridge
 - configuration
 - checking 97
 - prerequisites for 93
 - statistics, displaying 99
 - troubleshooting 99
 - learning network configurations 125
 - overview 111
 - security
 - combining source and destination features 106
 - destination explicit blocking, configuring 105
 - destination explicit forwarding, configuring 104
 - restricting packet forwarding and blocking 101
 - source explicit blocking, configuring 103
 - source explicit forwarding, configuring 102
 - standard filtering 125
 - transparent bridge routing table 124
 - bridge filtering examples 1233
 - bridge/router
 - accessing
 - via remote mode 1295
 - via Rlogin 1143
 - configuring
 - for bridging and routing 96
 - for incoming tunnel connection requests 742
 - bridge routing table, static entries 101
 - BRidge Service statistics 1367
 - BridgeNumber parameter 141
 - bridging
 - See also* source route bridging
 - basic 93 to 97
 - CONTRol parameter 101
 - customizing 101
 - DStSecurity parameter 102
 - filtering standard 107
 - firewalls 96
 - learning 125
 - load sharing 124
 - mapping
 - adding functional address to multicast address 109
 - address 115
 - user and access priorities 115
 - over MLN 88, 94, 101
 - packets
 - LLC length 115
 - MTU size on LANs 115
 - routes
 - displaying 124
 - dynamic 124
 - static 101
 - security 101 to 107
 - Spanning Tree Algorithm
 - and local area bridges 117
 - and wide area bridges 121
 - designated bridge 121
 - domain 123

network topology
 reconfiguration 124
 parameters, modifying 123
 prerequisites for
 configuration 118
 root bridge 119
 root port 119, 120
 structure 117
 SRcSecurity parameter 102
 statistics gathering 99, 1367, 1403
 translation
 between Ethernet and token ring
 networks 114
 configuring 108
 protocol support 108
 restrictions for AppleTalk 116
 restrictions for IPX 117
 transparent
 address format 110
 description 111
 enabling 101
 over ATM 1179
 over ATM DXI 1215
 over Frame Relay 1012
 over MLN 94, 101
 over SMDS 1049
 over WANs 93
 over X.25 1099
 per port 101
 setting up 93
 Broadcast and Unknown Server. *See* BUS
 BroadCastAddr parameter 809
 BroadcastLimit parameter 141
 BSC
 baud rates supported for 792, 800
 BscCU parameter 793, 794, 801
 configuration examples 795, 801
 configuring
 for central sites 793
 for remote sites 792, 800
 pass-through 791, 799
 CONTRol parameter 793, 794, 801
 CUCONTRol parameter 795
 defining CUs 794
 defining primary and secondary
 devices 793, 794, 801
 protocols supported 791
 Role parameter 793, 794, 801
 BSC Service statistics 1368
 BTU size 717
 bulletin board service 1496
 BUS 1210

C

cable length, external devices 1438
 CacheTime parameter 350
 caching, macro 1356
 Calling Line Identification Presentation. *See* CLIP
 CCITT Simple Standard PAD Profiles 1436
 CHAP, authentication 914
 CircuitBal parameter 751
 CIRcuits parameter 737
 CLIP
 dial staring 988
 identifying incoming calls
 overriding SCID configuration 74
 port configuration 989

remote site identification 988
 setting up with Port Service
 parameters 74
 CLNP for OSI routing
 displaying End System table 674
 displaying Intermediate System
 table 674
 enabling 658
 parameters for generating PDUs 675
 CLNP Service statistics 1369
 clocking, serial lines 1438
 COMmunity parameter 1286
 compression statistics 1271
 compression, data. *See* data compression
 ConfigCOS parameter 550
 configuration files 886, 890
 CONFIguration parameter
 AppleTalk 620
 APPN 498
 configuration statistics 618
 configuring
 CUs for SDLC 717
 MAC/SAP, SDLC devices 715
 remote SDLC devices 713
 SAP for the CU 718
 secondary SDLC devices 713
 wide area networks 496
 configuring NAT 319
 configuring the LLC2 data link
 interface 705
 congestion control
 configuring 1008
 Connect command 1159
 Connection 511
 CONNECTION parameter 520
 Connectionless Network Protocol. *See* CLNP
 connections
 displaying information for 520
 incoming. *See* incoming connections
 outgoing. *See* outgoing connections
 CONNECTIONS parameter 737
 ConnectionUsage parameter 706
 ConnNetworkChar parameter 513
 ConnNetworkDef parameter 513
 CONTRol parameter
 AppleTalk 633
 APPN 493
 ARP 225
 BRidge 101
 BSC 793, 794, 801
 DECnet 642
 DLsw 734, 767
 IDP 699
 IP 200
 IPV6 315
 LNM 1301
 RDP 379
 RIPXNS 699
 scheduling events 1305
 SNA Service 788
 control structures 1354 to 1355
 conventions
 notice icons, About This Guide 60
 text, About This Guide 60
 COSDef parameter 551
 COSNodeRow parameter 550
 COST parameter 642
 Cost parameter 214
 cost, route

DECnet 642
 OSPF 214
 COSTgRow parameter 551
 CU operating mode 718
 CUADDRess parameter 809
 CUCONTRol parameter
 ATUN Service 811
 BSC Service 795

D

data compression
 Boundary Routing environment 859
 choosing tinygram or link-level 1273
 configuring 1269
 link-level 1269 to 1272
 LinkCompStat parameter 1274
 operation 1272
 tinygram (packet-level)
 configuring 1269
 description 1272
 Data Link Connection Identifier. *See* DLCI
 data link switching
 circuit balancing 749
 configuring between APPN
 nodes 508
 connections 744 to 762
 converting SNA alerts to traps 759
 customizing 741
 displaying end-station topology 739
 for NetBIOS 734
 for SNA 731
 local switching port groups
 configuring 751
 deleting 755
 log display 738
 multicast 765
 configuring for NetBIOS mesh
 environments 766
 configuring for SNA client and
 server environments 767
 disabling 770
 restoring the default multicast
 address 769
 tuning parameters 769
 non-secure host configuration 741
 prioritizing traffic 745
 security access filter
 for NetBIOS traffic 744
 for SNA traffic 743
 setting bandwidth allocations and
 priorities 747
 source route dual-TIC topologies 758
 Spanning Tree Protocol (STP) 762
 terms 763
 tracing DLsw packets 1455
 Data Link Switching protocol. *See* DLsw
 data prioritization. *See* prioritizing data
 data rate, serial lines 1438
 data service unit. *See* DSU
 DataBits parameter 807
 daytime service, mapping to UDP
 ports 387
 decapsulation, X.25 switching 1116
 DECnet routing
 AdvertisePolicy parameter 641
 AdvToNeighbor parameter 642
 area to pseudo areas
 translation 651 to 652

- configuration 639
 - configuring
 - over Frame Relay 1019
 - over LANs 637
 - over SMDS 1056
 - over X.25 1085
 - CONTRol parameter 642
 - COST parameter 642
 - description 643
 - end nodes 643
 - filtering, setting up 641
 - HelloTime parameter 643
 - LAN Address Administration
 - restrictions 784
 - network
 - operations on 643
 - reachability 646
 - packets
 - forwarding 644
 - hello, transmission interval 643
 - triggered update 642
 - update 646
 - update, transmission interval 642
 - Phase IV to Phase V
 - terminology 654
 - transition sample
 - configuration 653
 - translation 650
 - PolicyControl parameter 642
 - PRIOrity parameter 642, 643
 - pseudo area configuration 652
 - RcvFromNeighbor parameter 642
 - ReceivePolicy parameter 641
 - router priority on LANs 642
 - routes
 - aging 643
 - learning 646
 - least cost 647
 - setting cost for 642
 - RoutingTime parameter 642, 646
 - split horizon 646
 - statistics display 1370
 - WAN configurations 639
 - DECnet Service statistics 1370
 - Default port owner
 - for WAN ports 75
 - default router, RDP Service 378, 381
 - DefaultMetric parameter 200
 - DefaultPriority parameter 1281
 - DefaultPU parameter 788
 - DefaultTTL parameter 226
 - DefaultZone parameter 615, 617
 - DEFine command 1139
 - defining CUs for SDLC 716
 - destination explicit blocking (DEB) 105
 - destination explicit forwarding (DEF) 104
 - DHCP
 - authorized server list 392
 - description 396
 - port numbers 388
 - relay agent 396
 - relaying BOOTP and DHCP traffic 391
 - Dial command 971
 - dial-on-demand
 - See also* dial-up lines
 - configuring 984
 - description 968
 - dial pool, definition 1005, 1006
 - dial-up lines
 - bandwidth allocation 969
 - bandwidth-on-demand 862
 - configuring 984
 - description 969
 - BODIncrLimit parameter 969
 - BODTHreshold parameter 969
 - checking path status 972
 - configuring 978
 - Data Terminal Ready (DTR) signal 966
 - Dial command 971
 - dial number list
 - editing 987
 - using 985 to 987
 - dial-on-demand
 - configuring 984
 - description 968
 - IPX in Boundary Routing
 - environment 1000
 - NCP connection process 589
 - NCP spoofing
 - configurations 590 to 1001
 - over IP network 997
 - over IPX network 999
 - over RIIP network 998
 - SPX1 watchdog packets on 589
 - type of routed packets 969
 - dial pool
 - DTR dialing and path
 - preference 970
 - leased line and path
 - preference 971
 - mapping remote caller ID to 990
 - path preference 970, 972, 989
 - DialIdleTime parameter 969
 - DialInitState parameter 968, 971
 - DialNoList parameter 968, 970
 - DialSamplPeriod parameter 969
 - DialStatus parameter 972
 - disaster recovery 862, 970
 - dynamic physical path 968
 - dynamic WAN Extender virtual
 - path 968
 - E1 line, configuring 982
 - HangUp command 971, 972
 - ISDN line 965
 - leased line 965, 968
 - MlpCONTRol parameter 995, 1005
 - modem pooling 967
 - NORMalBandwidth parameter 969
 - parameters and commands 1001
 - PathPreference parameter 970, 971
 - paths
 - dynamic, definition 1006
 - dynamic, description 966
 - static, definition 1006
 - static, description 966
 - phone list 968
 - port-based dialing
 - configuring 992
 - definition 971
 - port-based disconnecting
 - configuring 992
 - how to 972
 - PPP virtual ports 968
 - remote system's caller ID 967
 - static dial path 968
 - Switched-56 line
 - configuring 982
 - definition 965
 - T1 line
 - configuring 982
 - definition 965
 - T3 line
 - configuring 982
 - definition 965
 - telephone line
 - configuring 978
 - definition 965
 - terms 1005
 - troubleshooting configuration 994
 - verifying configuration 993
 - dial-up service commands 1001
 - DialIdleTime parameter 969
 - DialInitState parameter 968, 971
 - DialNoList parameter 928, 968, 970
 - DialSamplPeriod parameter 969
 - dial-up options, WAN Extender 943
 - directory information (APPN) 518
 - DIRectory parameter 518
 - DirectoryEntry parameter 503
 - DirectoryManage command 1146
 - disaster recovery
 - configuring over Frame Relay 1028
 - using virtual ports 1039
 - DisConnect command 1173
 - DiscoverRoutes command 144
 - DiscRouteRs command 379
 - Distance Vector Multicast Routing Protocol.
 - See* IP multicasting
 - DLCI
 - and dynamic configuration 1020
 - number assignment 1038
 - number for SNA traffic 865
 - DLSw
 - definition 725
 - tunnels 729
 - DLSw multicast. *See* multicast data link switching
 - DLSw Service statistics 1371
 - DLSw sessions with SDLC 722
 - DLSw. *See* data link switching
 - DlswLOG parameter 738
 - DlurDefaults parameter 491
 - DlurLinkSta parameter 492, 497
 - DluRStatus parameter 497
 - DluSStatus parameter 497
 - domain name service
 - for TCP/IP connections 1141, 1142
 - mapping to UDP ports 387
 - DownStreamLU parameter 497
 - DPM statistics 1404
 - DStSecurity parameter 102
 - DSU, ATM DXI 1213
 - dual PVC, configuring for SNA traffic 865
 - DVMRP Service statistics 1373
 - Dynamic Host Configuration Protocol. *See* DHCP
 - dynamic paths 64
-
- ## E
- E1 channelized leased lines 74
 - E1 lines
 - configuring 982
 - definition 965
 - E3 lines
 - configuring 982
 - definition 965
 - enabling NAT ports 319
 - encapsulation type

ATM 1188
 ATM DXI 1215, 1216
 Ethernet 802.2 to and from token ring
 802.2 152
 Frame Relay 1215, 1216
 LLC/SNAP 1188, 1215, 1216
 LLC-based token ring to and from
 Ethernet II 153
 NLPID 1215, 1216
 null 1188
 X.25 switching 1107, 1116
 encryption devices 1298
 end nodes (APPN)
 defining as adjacent node 503
 definition 525
 end system configurations
 IP security options 443
 route discovery 142
 End System Hello (ESH) packets 674
 End System to Intermediate System
 Protocol. *See* ESIS
 Entity Filters 624
 EntityFilter parameter 626
 EntityFilterNum parameter 626
 error messages
 for failed connections 1166
 ICMP 450
 ESIS for OSI routing
 configuration parameters 674
 enabling 658
 ESP header 405
 extended connections, incoming. *See*
 incoming connections and sessions
 extended connections, outgoing. *See*
 outgoing connections
 Exterior Gateway Protocol. *See* EGP
 ExteriorPolicy parameter 200
 external devices, serial lines 1437

F

fax service (3ComFacts) 1496
 FDDI
 port configuration 1217
 troubleshooting 1217 to 1218
 FEP 729
 filtering
 actions 1227 to 1229
 AppleTalk 622 to 626
 bridge examples 1233 to 1241
 bridging 107
 built-in masks 1229
 configuring filters 1221
 DECnet 641
 description 1225
 IBM traces
 DLSw packets 1455
 LLC2 frames 1459
 SDLC frames 1462
 IP 173
 IPX examples 1241 to 1245
 MASK parameter 1221, 1227
 parameter list 1225
 POLicy parameter 1221, 1227
 protocol reservation
 IP filtering procedure 178
 mnemonic filtering
 procedure 1228
 qualification 1227

user-defined masks 578, 1232
 firewalls
 conceptual information 417
 configuration
 blocking unwanted traffic 402
 defining a stance 398
 OAM procedures 399
 routing functions 399
 verifying 400
 filters
 IP versus firewall 413
 managing 410
 types 418
 FTP, managing connections 419
 setting up logs 414
 terminology 420
 FORwards command 1172
 ForwardTable parameter
 DVMRP 351
 FR Service statistics 1374
 Frame Relay
 addresses
 DLCI 1020, 1037
 example 1038
 AppleTalk routing 1015
 configuring
 APPN 1016
 bridge/router 1007
 data transmittal and
 retrieval 1008
 DECnet routing 1019
 disaster recovery 1028
 dual PVCS for SNA traffic 865,
 1028
 for Boundary Routing 822
 IP routing 1020
 IPX routing 1023, 1215
 OSI routing 1025
 source route bridging 1013,
 1180
 transmit network data 1008
 transparent bridging 1012
 verification 1012
 VINES routing 1026
 virtual ports 1018
 XNS routing 1027
 configuring congestion control 1008
 encapsulation type 1215, 1216
 Local Management Interface (LMI)
 Protocol 1039
 routing protocols supported 1013
 setting up 1007
 statistics display 1374
 topologies
 fully meshed 1034
 fully redundant 1041
 nonmeshed 1035
 partially meshed 1036
 partially redundant 1040
 transparent bridging over 94
 Frame Relay Access Device (FRAD)
 address mappings
 configuring 775, 776
 capabilities 771
 configuring
 FRAD node 771
 LAN-attached end stations 771,
 773
 SDLC-attached end stations 774,
 775

FrameChars parameter 807
 FrameGap parameter 808
 FrameSize parameter 807
 FrToAtm utility 1214
 FTP
 using System IP 243

G

global switching. *See* local and global
 switching
 Government Open Systems Interconnection
 Profile (GOSIP) 1339
 GREP, command examples 1433
 group 1067
 group ports. *See* multiple logical networks

H

HangUp command 971
 HDLC 725
 tunneling
 configuring 725
 prerequisites for 725
 typical uses 729
 HelloTime parameter 211, 643
 high-level data link control. *See* HDLC
 HoldTime parameter 147
 host name service, mapping to UDP
 ports 387
 HOSTS2 name service, mapping to UDP
 ports 388
 hot swapping hardware modules 1311
 HprTimer parameter 540
 HSS port, utilization percentage 1437

I

I/O module, token ring 1437
 IBM
 APPN references 534
 Boundary Routing. *See* Boundary
 Routing
 class of service mode defaults
 (APPN) 549
 trace facility 1455
 DLSw packets 1455
 LLC2 frames 1459
 SDLC frames 1462
 IBM bridge connectivity to 3Com token
 ring bridges 138
 ICMP error messages 450
 ICMP Redirect message 380
 ICMP Router Advertisement message. *See*
 also RDP Service. 380, 381
 ICMP Router Discovery Protocol. *See* RDP
 Service.
 ICMP Router Solicitation message. *See also*
 RDP Service. 380, 381
 ICMPGenerate parameter 226
 ICMPReply parameter 226
 IdleTimer parameter 807
 IDP for XNS routing
 CONTrol parameter values 699
 displaying
 statistics 696
 XNS Routing Table 701
 enabling 693
 IDP Service statistics 1374

- using RIP 203
- BGP
 - AS-path permit or deny policies 193
 - AS-path weight policies 195
 - default route 187
 - DefaultMetric parameter 191
 - degree of preference calculation 196
 - deny filter examples 194
 - ExteriorPolicy parameter 192
 - importing routes from BGP 191
 - importing routes from IGP 189
 - multi-homed autonomous systems 191
 - network number policies 192
 - peers 186
 - permit filter examples 194
 - regular expressions examples 194
 - route aggregation 188, 225
 - stub autonomous systems 191
 - transit autonomous systems 192
 - weight filter examples 196
- configuration, adding a dynamic address map 321
- configuration, checking
 - displaying statistics 162, 321
 - examining network devices 160
 - overall status 162
 - tracing routes 163
 - using PING command 160
- configuring
 - multiple IP subnets 163
 - over ATM 1181
 - over Frame Relay 1020
 - over LANs 157 to 159
 - over MLN 164
 - over SMDS 1057
 - over WANs 159
 - over X.25 1086
 - PPP 157 to 159
- CONTRol parameter 200
- Cost parameter 214
- customizing 163
- default routes 201
- DefaultMetric parameter 200
- DefaultTTL parameter 226
- DemandInterface parameter 212
- ExteriorPolicy parameter 200
- filtering
 - configuration examples 175
 - configuring 173
 - filter policy, setting up 173
 - setting up protocol reservation with IP filtering 178
- global configurations 226
- HelloTime parameter 211
- ICMPGenerate parameter 226
- ICMPReply parameter 226
- IISIS
 - configuring for dual IP and OSI mode 215
 - routing policies 184, 185
- InteriorPolicy parameter 200
- link state advertisement (LSA) 213
- load splitting 200
- LocalAS parameter 186
- logical network configuration 164
- MLN configuration 164
- multipath routing 199
- multiple logical networks 164
- NETaddr parameter 158
- network
 - reachability 204, 210
 - topology 197
- OSPF
 - configuration parameters 214
 - demand interface circuits 212
 - route cost 214
 - routing policies 182
- packets
 - broadcast 203
 - ICMP generation 226
 - ICMP reply 226
 - OSPF hello 210
 - RIP update 204, 206
- PeerAS parameter 186
- peers, internal and external BGP 216
- prerequisites 157
- ReassemblyTime parameter 226
- ReceivePolicy parameter 179
- RIP routing policies 179
- RIPIP parameters for RIP updates 206
- RIP-learned route states 207
- route selection 200, 201
- router
 - adjacencies 211
 - operations 196
 - security. *See* IP security
- ROUTerPriority parameter 210
- routes
 - BGP aggregation 188, 225
 - costs, reducing with demand circuits 158
 - costs, reducing with demand interface circuits 212
 - default 201
 - importing 202
 - learning with OSPF 208, 210
 - learning with RIP 203
 - RIPIP aggregation 166
 - selecting least cost 199
 - static 171
- running unnumbered links 159
- split horizon
 - next-hop 205
 - solving slow convergence 204
 - with poison reverse 206
- static routes 171, 172
- StaticPolicy parameter 200
- statistics display
 - ARP Service 1363
 - BGP Service 1366
 - OSPF Service 1390
 - RIPIP Service 1396
 - TCP Service 1404
 - UDPHELP Service 1406
- statistics gathering 162
- TraceRoute command 163
- UDP Broadcast Helper 163
- UpdateTime parameter 200
- variable length subnet masks
 - aggregation with RIPIP 166
 - range table mask with RIPIP 168, 169, 170
- WAN configurations 159
- IP security
 - attacks, preventing
 - filtering router 451
 - firewalls 455
 - multiple contiguous IP networks 454
 - multiple subnets 453
 - noncontiguous IP networks 452
 - routers from other vendors 454
 - attacks, types of 450, 451
 - configuration
 - checking 450
 - prerequisites for 443, 445
 - configuring
 - extended security option labels 449
 - for end systems 443, 444
 - for IP routers 444
 - description 443
 - enabling security options 448
 - ICMP error messages 450
 - port configuration
 - examples 446 to 448
 - terminology 455
- IP Security Protocol. *See* IPSec.
- IPSec
 - authentication header 463
 - configuring dynamic security policies 460
 - configuring IPSec 457
 - configuring IPSec with manual policy 460
 - configuring manual key information 459
 - configuring manual security policies 458
 - creating manual policies 457
 - creating manual policy key sets 458
 - dynamic policy customized security associations 461
 - dynamic policy IKEProfile 461
 - dynamic policy PreSharedKey 461
 - dynamic policy selector lists 461
 - dynamic policy transform lists 461
 - DynamicPolicy parameter 461
 - enabling 462
 - encapsulation security payload 462
 - how IPSec works 462
 - IPSec control and PORT service control 457
 - overview 457
 - policies 462
 - sample configurations 463 to 473
 - transport mode 457
 - tunnel mode 457
 - tunneling protocols 457
- IPv6 routing
 - autonomous systems
 - reducing network overhead 315
 - configuration, checking
 - displaying statistics 309
 - examining network devices 308
 - overall status 309
 - configuring
 - over LANs 307 to 308
 - PPP 307 to 308
 - CONTRol parameter 315
 - customizing 310
 - default routes 315
 - load splitting 315
 - multipath routing 314
 - NETaddr parameter 307
 - network
 - reachability 316

- topology 314
 - packets
 - broadcast 316
 - RIPNG update 316
 - prerequisites 307
 - ReceivePolicy parameter 312
 - RIPNG learned route states 317, 318
 - RIPNG routing policies 312
 - route selection 315
 - router
 - operations 314
 - routes
 - default 315
 - importing 316
 - learning with RIPNG 316
 - selecting least cost 314
 - static 311
 - running unnumbered links 308
 - split horizon
 - with poison reverse 317
 - static routes 311
 - UpdateTime parameter 315
 - WAN configurations 308
 - IPX filtering
 - examples 1241
 - forwarding/discarding packets 1221
 - IPX routing
 - AdvertisePolicy parameter 605
 - AdvToNeighbor parameter 584, 607
 - AllServers parameter 598, 601
 - configuration
 - checking 570
 - displaying statistics 572
 - examples 586
 - troubleshooting 572
 - configuring
 - for NLSP 569
 - IPXWAN over PPP 567
 - neighbors 584
 - over ATM 1184
 - over ATM DXI 1215
 - over Frame Relay 1023, 1215
 - over LANs 563
 - over SMDS 1060
 - secondary networks with different header formats 564
 - CONTrol parameter 575, 576
 - customizing 574
 - dial-on-demand 589 to 595
 - filtering
 - built-in masks 578
 - user-defined masks 578
 - header formats 564, 566
 - local and wide area network configuration 596
 - network reachability 601
 - packets
 - encapsulation format 564
 - triggered RIP updates 575
 - unknown destination 582
 - policies
 - deriving advertised routes from service policies 605
 - description 603
 - disabling 604
 - neighbor 604, 607
 - normal and inverse lists 585
 - Novell service types 608
 - overriding 604
 - RIP 585, 603
 - route advertisement 605
 - route receive 605
 - SAP 585, 604
 - service advertisement 607
 - service receive 606
 - PolicyControl parameter 584, 604
 - RcvFromNeighbor parameter 607
 - ReceivePolicy parameter 605
 - RIP and SAP updates
 - controlling 575
 - nonperiodic 576, 600
 - packet contents 601
 - periodic 576, 600
 - transmission interval 577
 - ROUte parameter 580
 - router 595, 596, 721
 - routes
 - aging, controlling 577
 - controlling advertisement of 575
 - default 582, 599
 - default metric 583, 599
 - dynamic learning, enabling and disabling 575
 - learning 600
 - selecting 599
 - static, adding 580
 - static, deleting 582
 - routing table
 - displaying 597
 - flushing dynamic routes 577
 - server table
 - displaying 598, 601
 - flushing 577
 - service
 - aging, controlling 577
 - information, learning 600
 - static servers, adding and deleting 584
 - split horizon
 - next-hop 566, 584, 596, 602
 - solving slow convergence 601
 - with poison reverse 576, 603
 - static servers, adding and deleting 579
 - statistics display
 - IPX Service 1381
 - NLSP Service 1387
 - NRIP Service 1389
 - SAP Service 1398
 - UpdateTime parameter 577
 - WAN configurations 566
 - IPX Service statistics 1381
 - IPX translation bridging restrictions 117
 - IPX25Map parameter 1125
 - IPXWAN, configuring over PPP 567
 - ISDN
 - addresses 925, 927
 - BRI 922
 - configuring
 - data rate transfer 924
 - dialup 919
 - remote device 922
 - configuring paths 72
 - deciding how to use interface 919
 - dialup. *See* dial-up lines
 - paths
 - numbering 67, 68
 - phantom power 922
 - planning network 918
 - ports
 - configuring 72
 - numbering 67, 68
 - products offered 918
 - TAs, recommended 918
 - topologies, common 919
 - virtual ports
 - configuring 84
 - numbering 67, 68
 - ISDN lines
 - configuring 980
 - configuring for SNA traffic over dial-up line 998
 - Service Profile Identifiers (SPIDs) 982
 - summary of dial-up commands and parameters 1001
 - support for 966
 - ISDN topology
 - boundary routing with disaster recovery 920
 - boundary routing with redundant routes for networks 920
 - ISDN as backup 920
 - traditional routed 920, 921
 - ISIS for OSI routing
 - configuration parameters 675
 - enabling 658
 - interdomain routing example 676
 - ISIS Service statistics 1382
 - ISRsessions parameter 497, 522
-
- L**
- L2Tunnel Service
 - access list configuration 290
 - virtual leased lines adding 293
 - virtual leased lines deleting 293
 - LAN Address Administration (LAA)
 - assigning a MAC address to a CEC interface 783
 - assigning a MAC address to a path 781
 - configuring with DECnet 784
 - resetting MAC address to default 782
 - LAN emulation 709, 1197, 1210
 - LAN Emulation Client. *See* LEC
 - LAN Emulation Configuration Server. *See* LECS
 - LAN Emulation Server. *See* LES
 - LAN Emulation User Network Interface. *See* LUNI
 - LAN Net Manager support 1299 to 1301
 - LAPB, configuring 913
 - LargestFrameSize parameter 139
 - leased lines
 - configuring 982
 - definition 965
 - LEC 1210
 - LECS 1210
 - left-hand side address. *See* LHS address
 - LEN end nodes
 - defining as adjacent nodes 503
 - definition 525
 - preconfiguring LUs in network node directory 501
 - registering LUs on 503
 - LES 1210
 - LHS address 322
 - LifeTime parameter 378

Link Access Procedure Balanced Mode. *See* LAPB

Link Control Protocol (LCP) packet, loopback detection using magic numbers 914

link state advertisement (LSA) 213

LinkCompStat parameter 1274

link-level compression 1269, 1273

LinkStaCHar parameter 489, 500

LinkStaCONT parameter 789

LinkStaCONTrol parameter 494, 517, 521

LLC/SNAP encapsulation 1188, 1215, 1216

LLC2 data link interface
configuring 705
tracing LLC2 frames 1459

LLC2 data link interface
configuration 705

LLC2 Service statistics 1384

LLC2 tunneling. *See* tunnel connections

LLC2-bridged packets, prioritizing 1278

load balancing
bandwidth-on-demand 915
bundle 916
dial path pooling 916
on PPP links 915, 994, 1005
sequencing 915

load sharing 330
bandwidth-on-demand 915
bundle 916
in bridges 124
on PPP links 915
sequencing 915

local access control
configuring 1155

local and global switching
configuring 1107 to 1109
X.25 prefix address mapping 1108

Local Management Interface (LMI) Protocol 1039, 1067

LocalDialNo parameter 925, 927

LocalMac parameter 810

LocalNodeName parameter
APPN Service 484
SNA Service 787

LocalNodeResist parameter 484

LocalSubAddr parameter 925

logfile, contents of 1289

logging messages, NAT Service 321

logging messages, RAS Service 441

Logging On and Logging Out 1156

logical networks. *See* multiple logical networks

logical ring in source route bridging 139

Logical Units (LUs)
deleting LEN end node LUs 503
registering LUs on LEN end nodes 503

LOGout command 1156

loopback testing
ATM connectivity 1178
HSS V.35 and RS-232
local test flowchart 1321
remote test flowchart 1323

ISDN, using B-channels 1324
local 1321
loopback fixture 1323
remote test 1319, 1322

LUNI 1210

LUNI Management Entity. *See* UME

M

MABR parameter 352

MAC addresses, assigning to a physical path 781

MacAddress parameter 781, 782

MacCache parameter 738

macros
caching and shared macros 1356
concatenated 1357
conditional statements 1349
control structures 1354
conventions 1349
creating and managing 1139
Event-Based Command/Macro Executor (EBME) 1308
example 1358
executing 1305, 1307
keywords 1355, 1356
larger macros 1357
memory considerations 1356
nesting in conditional statements 1358
port initialization with incoming automatic connections 1138

MANager parameter 1286

managing the network. *See* network management

manual dialing 971

many-to-many address mapping, NAT Service 320, 324

many-to-one address mapping, NAT Service 320, 324

mapping addresses
AppleTalk to Frame Relay DLCIs 1015
AppleTalk to SMDS individual address 1054
DECnet to Frame Relay DLCIs 1020
IP to Frame Relay DLCIs 1021
IP to X.25 1118, 1125
IPX to Frame Relay DLCIs 1024
OSI to Frame Relay DLCIs 1026
P-Selector to X.25 1125
VINES to Frame Relay DLCIs 1027
XNS to Frame Relay DLCIs 1028

mapping service names to UDP ports 387

MASK parameter 1221, 1227

MaxAreRDLimit parameter 142

MaxFrame parameter 705

MaxInterval parameter 378

MaxSteRDLimit parameter 142

MBONE. *See* IP multicasting

McastRetry parameter 769

McastTcpldle parameter 769

member ports. *See* multiple logical networks

menus, Add Name 1148

Meshed Topology with ISDN 919

MEtric parameter 347

MIB support 1345

MIBs 1495

migrating to a RIPv2 network 181

MinAccessPrior parameter 147

MInInterval parameter 379

MIP Service statistics 1386

MLN. *See* multiple logical networks

MLP, with load balancing 915, 1005

MlpCONTrol parameter 995, 1005

mnemonic 1228

mnemonic filtering. *See* filtering 125

MMode parameter 733, 766

Mode parameter 140

modems
DTR dialing 966, 970
loopback fixture 1323
loopback testing
local 1321
remote 1322
messages 1314
V.25 bis dialing 966

ModetoCosMap parameter 551

modules, hot swapping 1311

MOSPF Service statistics 1386

MRInfo command 338

MTraceRoute command 339

multicast data link switching 765
configuring
for NetBIOS mesh environments 766
for SNA client and server environments 767
disabling 770
restoring the default multicast address 769

Multicast Open Shortest Path First Protocol. *See* IP multicasting 335

MulticastAddr parameter 768

MultiLink Protocol. *See* MLP

multiple logical networks
bridging 94, 101
configuring port groups 91
description 87
external bridges 94
IP, configuring over 164
transparent bridging 94, 101

N

name service
for incoming OSI connections
file-based 1152
X.500 directory 1145
for incoming TCP/IP connections
domain 1141, 1142
IEN116 1141

NameCache parameter 738

names
entity, AppleTalk 630
path 71
port 71
SNMP community 1286

NAT 319
address mapping
dynamic 324
many-to-many 324
many-to-one 324
one-to-many 324
one-to-one 324
static 324
address migration 322, 330
address redirection 331
basic operation 322
configuring 319
guidelines for using 322
load sharing 322, 330
private address space 322, 327
session creation 323
session information 321
specifying session direction 323

- TCP port mapping 327
 - UDP port mapping 327
 - using a mask 325
 - NBBcastResend parameter 745
 - NBBcastTimeout parameter 745
 - NBRemAccess parameter 744
 - NCP
 - configuring 912
 - LCP connection with a RAS client 912
 - request packets supported 912
 - neighbor policy 603, 607
 - neighbor router, RDP Service 381
 - neighbors
 - IPX 584, 604, 607
 - OSPF 208
 - VINES 687
 - Neighbors parameter 1093
 - NETaddr parameter 158, 307
 - NetBIOS datagram service, mapping to UDP ports 388
 - NetBIOS name service, mapping to UDP ports 388
 - NETBuilder II
 - configuring ports and paths for local area interfaces 71
 - numbering ports and paths
 - configuration 64
 - on multiport hardware module 65
 - statically configured tables 1411
 - swapping hardware modules 1311
 - virtual ports
 - configuring for wide area
 - interfaces 84
 - functionality 77
 - inherited attributes 83
 - lack of connectivity 1037
 - over ATM 81
 - over Frame Relay, ATM DXI, and X.25 80
 - over PPP 82
 - over SMDS 82
 - NetFilter parameter 623
 - NetFilterType parameter 623
 - NetLogin prompt 1156
 - NetMapTime parameter 1288
 - NetRange parameter 615, 616
 - NetView run commands support 1483
 - NetView Service Point
 - activating and deactivating SSCP link stations 789
 - activating and deactivating SSCP-PU sessions 789
 - configuring 787
 - Netware Link Services Protocol, IPX
 - area addressing 569, 610
 - configuring 569
 - description 609
 - hierarchical topology 609
 - Network Address Translation. *See* NAT
 - network clock
 - synchronizing 1438
 - Network Control Protocol. *See* NCP.
 - Network Entity Title (NET) 668
 - network management
 - Adapter Management Protocol (AMP)
 - Discovery Responder 1302
 - multicast and functional addresses
 - defaults** 1302
 - displaying** 1302
 - network device discovery 1301
 - ATM UNI UME 1191
 - audit trail messages 1413
 - community names 1286
 - encryption devices
 - resynchronization feature 1298
 - LAN Net Manager support 1299
 - manager list 1286
 - MANager parameter 1286
 - NetMapTime parameter 1288
 - network maps 1288
 - remote access 1295 to 1298
 - RMON alarm agent 1287
 - set request 1287
 - SNMP 1285 to 1287
 - Telnet access, restricting 1298
 - traps 1287
 - network node topology information 519
 - network resiliency 840, 865
 - Network Service Access Point (NSAP)
 - addressing. *See* NSAP and PSAP
 - addressing
 - network supplier support 1496
 - Network Termination 1. *See* NT1
 - network topology information,
 - displaying 518
 - network traffic, reducing. *See* split horizon
 - NetZoneMapping parameter 617, 630
 - NIC host name service, mapping to UDP ports 388
 - NLPID encapsulation 1215, 1216
 - NLSP Service statistics 1387
 - NNtopology parameter 519
 - NORMalBandwidth parameter 969
 - Novell
 - interoperability for IPX over WANs 567
 - NetWare connectivity between IPX router 595, 721
 - NetWare packets spoofed over dial-on-demand lines 589
 - service types 608
 - NRIP Service statistics 1389
 - NSAP address
 - Phase IV 655
 - prefixes 678
 - structure 666
 - NSAP and PSAP addressing 1339 to 1343
 - NT1
 - definition 922
 - disabling phantom power 922
 - power sources 922
 - NumAltMgrs parameter 1300
 - number conventions for built-in ISDN interfaces 69
-
- O**
 - OfficeConnect NETBuilder
 - numbering ports and paths 68
 - one-to-many address mapping, NAT
 - Service 320, 324
 - one-to-one address mapping, NAT
 - Service 320, 324
 - online technical services 1495
 - OPING command 1168
 - OSI connections
 - incoming
 - checking network resources 1168
 - enabling 1159
 - file-based name service 1144
 - session management. *See* sessions
 - X.500 directory service 1145
 - outgoing 1122
 - OSI routing
 - area address
 - assigning 667
 - NSAP address structure 666
 - areas
 - description 668
 - single leaf 672
 - transit 672
 - changing level of routing 664
 - CLNP parameters 675
 - configuration
 - displaying statistics 662
 - troubleshooting 662
 - verifying 659
 - configuring
 - basic routing 657
 - for WANs 659
 - Integrated IS-IS for IP and dual IP/OSI mode. *See* IP routing
 - over Frame Relay 1025
 - over SMDS 1062
 - over X.25 1092
 - customizing 664
 - description 665
 - End System table 674
 - ESIS parameters 674
 - ESIS Protocol 657
 - interdomain routing
 - address extraction for X.25 and SMDS-based NSAPs 679
 - address prefix 678
 - configuring 676
 - Interdomain Routing Table 680
 - Intermediate System table 674
 - ISIS parameters 675
 - ISIS Protocol 657
 - interdomain routing
 - address extraction for X.25 and SMDS-based NSAPs 679
 - address prefix 678
 - configuring 676
 - Interdomain Routing Table 680
 - Intermediate System table 674
 - ISIS parameters 675
 - ISIS Protocol 657
 - Level 1 routing 665, 668
 - Level 1 Routing Table 669
 - Level 2 routing 665, 670
 - Level 2 Routing Table 672
 - load splitting 673
 - multipath routing 673
 - Network Entity Title (NET) 667
 - network topology 665
 - packets
 - End System Hello (ESH) 668, 674
 - Intermediate System Hello (ISH) 668, 674
 - route cost and selection 673
 - statistics, displaying 662, 1368, 1369, 1382
 - troubleshooting 660, 662
 - WAN configurations 659
 - OSI Virtual Terminal Protocol (VTP) 1117
 - OSPF
 - adjacencies 211
 - configuration parameters 214
 - Cost parameter 214
 - demand interface circuits 212
 - DemandInterface parameter 212
 - HelloTime parameter 211
 - learning routes 210
 - link state advertisement (LSA) 213
 - route cost 214
 - route policies 181

router functions 208
 ROUTerPriority parameter 210
 OSPF router ID
 using System IP 243
 OSPF Service statistics 1390
 outgoing connections
 addresses
 mapping IP to X.25 1125
 mapping P-Selector to X.25 1125
 automatic
 description 1131
 initiating 1126
 logging out 1126
 X.25 address strings 1119
 extended
 description 1131
 exiting from PAD mode
 prompt 1129
 initiating 1126
 PAD emulation mode 1121, 1126
 PAD parameters, modifying 1127
 virtual call, establishing 1127
 IPX25Map parameter 1125
 OSI Virtual Terminal Protocol (VTP) 1117
 overview 1130
 port selection 1126
 PSELX25Map parameter 1125
 Telnet
 configuration example 1118
 configuring X.25
 gateway 1117 to 1121
 troubleshooting 1129
 VTP (OSI)
 configuration example 1123
 configuring X.25
 gateway 1122 to 1125

P

packets
 broadcast
 RIP 203
 RIPNG 316
 route propagation frequency 621
 UDP Broadcast Helper 395
 destination explicit blocking 105
 destination explicit forwarding 104
 encapsulation format, IPX 564
 end system, route discovery for 144
 extended security option labels, IP 449
 filtering
 AppleTalk 622 to 626
 DECnet 641
 IP 173
 on a bridge 125, 1221
 forwarding
 ratio 1283
 restrictions 101
 hello
 DECnet 643
 ESH PDUs 674
 ISH PDUs 674
 OSPF 210
 IP fragmentation 117
 KeepAlive
 NCP spoofing over dial-on-demand lines 590

NetWare 589
 LCP 914
 prioritizing. *See* prioritizing data
 source explicit blocking 103
 source explicit forwarding 102
 source route transparent bridging gateway
 description 151
 SR-to-TB domain handling 151
 TB-to-SR domain handling 152
 spoofed NetWare 591, 593
 update
 DECnet 642, 646
 OSPF 208
 RIP 204, 206, 698
 RIPNG 316
 VINES 691
 watchdog
 SPX spoofing over dial-on-demand lines 593
 PAD emulation mode 1127
 PAD profiles 1436
 PAP
 authentication 914
 setting up and verifying 908
 parallel bridges in source routing 141
 parameters
 bandwidth management service 1001
 CLNP, for OSI routing 675
 data prioritization 1279
 dial-up service 1001
 ESIS, for OSI routing 674
 Filter Service 1225
 ISIS, for OSI routing 675
 OSPF configuration 214
 RIP routing policy, for IP 179
 RIPNG, for RIP updates 206
 RIPNG routing policy, for IPV6 312
 RIPXNS, for RIP updates 698
 parent ports 82
 PARity parameter 807
 passive bridging 139
 password
 and userid pair 908
 changing 1156
 PassWord parameter 1300
 PATH Service statistics 1391
 PathPreference parameter 970, 971
 paths
 adding to path preference list 991
 appending to path preference list 991
 configuring local and wide area interfaces 71, 72
 converting static to dynamic 987
 defining dial path preference list 989
 definition 64, 1006
 deleting from path preference list 992
 dynamic binding
 definition 1006
 dynamic binding to port 987
 dynamic dial pool 64, 1005, 1006
 ISDN line, configuring for dial-up line 980
 multiple, mapping to one port 69
 numbering
 on NETBuilder II 64
 on OfficeConnect NETBuilder 68
 on SuperStack II 67
 on multiport hardware module 65
 removing from dial pool 987
 static 64
 static binding to port 1006
 telephone line, configuring for dial-up line 978, 982
 virtual, definition 1006
 PathSwitch command 540
 PEer parameter 734, 744
 PeerMacAdd parameter 745
 PeerNName parameter 745
 permanent virtual circuit. *See* PVC
 PermVirCircuit parameter 1188
 PhantomPower parameter 922
 PING command 161, 1167
 Point-to-Point Protocol. *See* PPP
 POLicy parameter 1221, 1227
 PolicyControl parameter
 DECnet 642
 NRIP 584, 604
 SAP 584, 604
 polled asynchronous communications. *See* asynch communications
 port compression statistics 1271
 port groups. *See* multiple logical networks
 PORT Service statistics 1393
 port services 327
 port to path mapping for SDLC 719
 PortCONTRol parameter 808
 APPN Service 516
 ATUN Service 808
 PortCU parameter 809
 PortDef parameter 536
 APPN Service 484
 SNA Service 787
 PortGroup parameter 755
 ports
 configuring
 FDDI 1217
 for local and wide area interfaces 71, 72
 multiple logical networks 91
 virtual ports 84
 defining for APPN 484
 defining for APPN HPR 536
 definition 64, 1006
 dynamic binding to path 1006
 dynamically activating and deactivating for APPN 516
 group. *See* multiple logical networks member. *See* multiple logical networks numbering
 convention in SNMP 1345
 on multiport hardware module 65
 on NETBuilder II 64
 on OfficeConnect NETBuilder 68
 on SuperStack II 67
 packets
 default priority 1281
 forwarding ratio, displaying 1284
 forwarding ratio, setting 1283
 parent 82
 serial
 utilization percentage 1437
 V.35 HSS module placement 1437
 static binding to path 1006
 virtual
 configuring for wide area interfaces 84
 definition 77
 inherited attributes 83

- lack of connectivity 1037
- number supported per
 - platform 78
- over ATM 81
- over Frame Relay, ATM DXI, and X.25 80
- over PPP 82
- over SMDS 82
- platforms supported on 78

PPP

- configuring
 - for Boundary Routing 817
 - IPXWAN over 567
 - LAPB for noisy lines 913
- configuring NCPs 912
- enabling 908
- Link Control Protocol (LCP)
 - packet 914
- load balancing 915, 994, 1005
- load sharing 915
- loopback detection using magic numbers 914
- packet size negotiation 913
- serial lines, maintaining quality of 914
- Spanning Tree Protocol 122

PPP Service

- CHAP, selecting 435
- PAP
 - selecting 435

PPP Service statistics 1394

PrefixRoute parameter 1093

Presentation Service Access Point (PSAP) addressing. *See* NSAP and PSAP addressing

primary PVC, configuring 1030

prioritizing data

- advantages of 1275
- assigning packet priority 1281
- assigning traffic priorities 745 to 749
- configuring priority 1276
- DefaultPriority parameter 1281
- interleave factor 1276, 1283
- MASK parameter 1281
- packets
 - default priority 1281
 - forwarding ratio, displaying 1284
 - system-assigned priority 1280
- parameters 1279
- queue arbitration algorithm 1283
- QueuePriority parameter 1280, 1281
- queues 1281
- to IP packets 1279
- to LLC-, SNA, and NetBIOS packets 1277
- TUNnelPriority parameter 1280, 1281

PRIOrity parameter 642, 643

PRIOrityCRiteria parameter 747

PRIOritySTATistics parameter 748

private address space 327

protocol reservation 1005, 1247

configuring

- APPN-routed traffic 1261
- bridged packets 1252
- DLSw (tunnel endpoint) 1258
- IP-routed packets 1253
- IPX-routed traffic 1256
- LLC2 traffic for SNA boundary routing 1259
- mixed bridge traffic
 - example 1262

- mixed-routed packets
 - example 1264
 - virtual port example 1265
- IP filtering procedure
 - description 173
 - FTP example, destination address 178
 - IP FilterAddr parameter action
 - option, PROTOcolRsrv= 173
 - IP FilterAddr parameter syntax 173
 - Telnet and FTP example, different protocols 177
- mnemonic filtering procedure
 - bridging example, destination address 1239
 - bridging example, different protocols 1240
 - bridging example, packets of specified lengths 1240
 - description 1228
 - Filter POLicy action option, PROTOcolRsrv 1228
 - IPX example, packet size 1245
 - procedural overview 1249
 - why to use 1247
- ProtocolRsrv parameter 1005
- PSelX25Map parameter 1125
- PVC
 - configuring 1010
 - configuring backup 1030
 - configuring dual circuits for SNA traffic 865, 1028
 - configuring primary 1030
 - definition 1210
 - setting up on X.25 1102

Q

- QueryInterval parameter 340
- queue arbitration algorithm 1283
- queue types 1281
- QueueInterLeave parameter 1283
- QueuePATtern parameter 1284
- QueuePriority parameter
 - APPN 1280, 1281
 - APPN Service 1281
 - IP 1280, 1281
 - LLC2 1280, 1281

R

- RateAdaption parameter 924
- RateLimit parameter 347
- RcvFromNeighbor parameter
 - DECnet 642
 - NRIP 607
 - SAP 607
- RcvSubnetMask parameter 168
- RDP Service
 - configuring 377
 - default router 378, 381
 - disabling 379
 - discovering neighboring RDP routers 379
 - discovery process 380
 - enabling 379
 - IP broadcasted packets 379
 - LifeTime parameter 378
- MAXInterval parameter 378
- message 378
- MinInterval parameter 379
- multicast packets 379
- neighboring router 380, 381
- participating routers 378
- router advertisement message 380, 381
- router solicitation message 380, 381
- RouterList parameter 378
- timers 378
- troubleshooting 380
- verifying configuration 380
- ReassemblyTime parameter 226
- ReceivePolicy parameter
 - DECnet 641
 - NRIP 605
 - RIPIP 179
 - RIPNG 312
 - SAP 606
- ReceiveWindow parameter 706
- record type 1413, 1417
- redundancy in Boundary Routing 840, 865
- regular expressions
 - AS filter examples 1433
 - components 1431
 - defined 1431
 - GREP command examples 1433
- remote addressing for SDLC 719
- Remote Boot and Configuration Services (RBCS) audit trail messages 1413
- REMOte command 1156, 1295
- Remote Network Monitoring (RMON)
 - alarms 1287
- remote SDLC devices, configuring 713
- remote site identification
 - with SCID and CLIP 988
- remote site identification options
 - WAN Extender 944
- RemoteMac parameter 810
- resiliency, network 840, 865
- RESume command 1172
- RetryCount parameter 705
- returning products for repair 1498
- RHS address 322
- RIF 149
- right-hand side address. *See* RHS address
- RIL 149
- RIP
 - for IP routing
 - changing states of routes 207
 - learning routes 203
 - range table mask for subnetting 168
 - RIPIP parameters for updates 206
 - route aggregation/deaggregation for subnetting 166
 - route policies 178
 - variable length subnet masks 166
- for IPX routing
 - periodic and nonperiodic updates 576
 - route policies 585, 603
 - triggered updates 575
- for XNS routing
 - CONTRol parameter values 699
 - displaying statistics 696
 - RIPXNS parameters for updates 698

- RIP policy 178, 585, 603, 605
 - RIPIP Service statistics 1396
 - RIPNG
 - for IPV6 routing
 - changing states of routes 317
 - learning routes 316
 - route policies 311
 - RIPNG policy 311
 - RIPXNS Service statistics 1397
 - Rlogin
 - connections
 - configuring 1143
 - to resources 1163
 - sessions. *See* sessions
 - RLOGin command 1143, 1163
 - route descriptor. *See* source route bridging, route designator
 - route discovery
 - All Routes Explorer frame 154
 - configuring per port 142
 - for end system source routing 155
 - Spanning Tree Explorer frame 155
 - ROUte parameter 144, 145, 580, 698
 - RouteAgingTime parameter 621
 - RouteDiscovery parameter 143, 144
 - Router Discovery Protocol (RDP). *See* RDP Service.
 - RouterList parameter 378
 - ROUTerPriority parameter 210
 - routes
 - aggregation
 - BGP 188, 225
 - DVMRP 348
 - RIPIP 166
 - cost
 - DECnet 647
 - OSPF 212, 214
 - default
 - BGP 187
 - ISIS 191, 202
 - OSPF 191, 202, 214
 - RIPIP 191, 202
 - demand circuits 212
 - importing from IGP to BGP domain 191
 - learning
 - AppleTalk 621
 - bridge 125
 - DECnet 646
 - IP, with BGP 216
 - IP, with OSPF 208
 - IP, with RIP 203
 - IP, within autonomous systems 214
 - IPV6, with RIPNG 316
 - IPX, with RIP 575
 - VINES 691
 - XNS 701
 - static
 - IP 171
 - IPV6 311
 - IPX 580
 - XNS 698
 - RouteTable parameter 350
 - RouteUpdateTime parameter 621
 - routing
 - AppleTalk over
 - LANs 614
 - non-AppleTalk data link 620
 - DECnet 637
 - IP 157
 - IP over ATM 1181
 - IP V6 307
 - IPX 564
 - IPX over ATM 1184
 - IPX over ATM DXI 1215
 - OSI 657
 - over
 - Frame Relay 1013
 - PPP 907
 - SMDS 1052
 - X.25 1081
 - VINES 683
 - XNS 693
 - XNS over ATM DXI 1215
 - routing informatin field. *See* RIF
 - routing information indicator. *See* RII
 - Routing Information Protocol Next Generation. *See* RIP
 - Routing Information Protocol. *See* RIP
 - Routing Table Maintenance Protocol (RTMP) 636
 - routing tables
 - Level 1 and 2 644
 - static
 - maximum entries allowed 1411
 - RoutingTime parameter 642, 646
 - RSVP
 - configuration example 274
 - configuration with L2TP tunnel 276
 - overview 273
 - proxy sender and receiver 274
 - rsvp
 - configuring 273
 - RSVP STATistics command 1397
 - RTP parameter 540
 - RTPStats parameter 540
-
- S**
- SampleTime parameter 618, 685, 696
 - SAP for IPX routing
 - periodic and nonperiodic updates 576
 - service policies 585, 604
 - SAP numbers
 - for SNA traffic on Frame Relay 865
 - SAP policy 585, 603, 605
 - SAP Service statistics 1398
 - scheduling events
 - CONTRol parameter 1305
 - creating 1305
 - scheduler 1305, 1307 to 1308
 - WAN dial-up connections 1306
 - Scheduling Service, macro
 - execution 1306
 - SCID
 - port configuration 988
 - remote site identification 988
 - SDLC
 - address mapping 722
 - configuring
 - clocking and line parameters 715
 - communication mode 715
 - connected devices 716
 - port mode for connected devices 716
 - port role 716
 - port timing 717
 - tranmission encoding 715
 - verification 719
 - connection methods 713
 - connectivity 721
 - conversion 721
 - definition 725
 - device mapping 721
 - devices 713
 - disabling LAPB 715
 - initiating sessions 724
 - mapping connections 721
 - polling 721
 - secondary device configuration 713
 - setting up for auto startup
 - tracing SDLC frames 1462
 - tunneling
 - configuring 725
 - prerequisites for 725
 - typical uses 729
 - SDLC Service, Boundary Routing 856
 - SdlcAdjLinkSta parameter 488, 499, 520, 537
 - SdlcDiurLinkSta parameter 492
 - SdlcLinkSta parameter 788
 - security
 - hijacked connections 451
 - IP attacks, preventing with route filtering 451
 - IP attacks, secure configurations
 - firewalls 455
 - multiple contiguous IP networks 454
 - multiple subnets 453
 - noncontiguous IP networks 452
 - routers from other vendors 454
 - IP spoofing 450
 - IP, security options feature 443
 - PPP 908
 - vulnerable configurations 451
 - seed router, AppleTalk 629
 - serial lines
 - clocking 1438
 - connectivity 1437
 - Frame Relay 1032
 - SMDS 1047
 - X.25 1073
 - managing 914
 - PPP 907
 - running PPP as unnumbered link 159
 - throughput, enhancing 1269
 - service
 - RSVP 273
 - Service Advertisement Protocol. *See* SAP
 - session management commands 1169
 - sessions
 - BACKwards command 1172
 - Connect command 1159
 - current
 - changing 1171
 - resuming 1172
 - DisConnect command 1173
 - displaying 1171, 1172
 - ECM character 1170, 1172
 - error messages 1166
 - FORwards command 1172
 - link and data link switching 722
 - managing 1169
 - multiple
 - connecting 1170
 - disconnecting 1173
 - network resources, checking

- OSI 1168
 - TCP/IP 1167
 - port modes 1170
 - RESume command 1172
 - resuming 1172
 - RLOGin command 1163
 - Rlogin connections 1163
 - single
 - connecting 1169
 - disconnecting 1173
 - TCP/IP and OSI connections 1159
 - TELnet command 1161
 - Telnet connections to TCP/IP
 - resources 1161
 - troubleshooting 1166
- SESSions parameter 706
- setting up 624, 626
- Setting Up a Permanent Virtual Circuit
 - Connection over X.25 1102
- SFTP service, mapping to UDP ports 388
- Simple Network Management Protocol (SNMP). *See* SNMP and network management
- slow convergence, solving. *See* split horizon
- smart filtering 856
- SMDS
 - addresses, group and individual 1067
 - AppleTalk route filtering 1071
 - basic bridging 1049
 - configuration 1048, 1049
 - configuring
 - AppleTalk routing 1052
 - data transmittal and retrieval 1048
 - DECnet routing 1056
 - DVMRP or MOSPF routing 342
 - IP routing 1057, 1071
 - IPX routing 1060
 - OSI routing 1062
 - routing protocols 1068
 - source route bridging 1051
 - transparent bridging 1049
 - VINES routing 1064
 - XNS routing 1065
 - description 1066
 - Local Management Interface (LMI) Protocol 1067
 - SMDS Interface Protocol (SIP) 1066
 - transparent bridging over 94
 - WAN 1047
- SMDS Service statistics 1400
- SNA
 - MAC addresses, assigning to physical paths 781
 - prioritizing NetBIOS-bridged packets 1278
 - traffic on Frame Relay 865
- SnaAlertsToTraps parameter 759
- SnaRemAccess parameter 743
- SnaTopoCollect parameter 739
- SnaTopoDisplay parameter 739
- SNI 1067
- SNMP
 - configuring 1286
 - description 1285
 - port numbering convention 1345
 - trap messages, audit trail notification 1295
 - using System IP 243
- SNMP Service statistics 1401
- source explicit blocking (SEB) 103
- source explicit forwarding (SEF) 102
- source route bridging
 - 3Com token ring and IBM bridge connectivity 138
 - AllRoutes parameter 146
 - basic 127
 - BLimitTimer parameter 141
 - BridgeNumber parameter 141
 - BroadcastLimit parameter 141
 - configuring
 - over WANs 130
 - source route bridging 127, 135
 - source route transparent bridging 135
 - source route transparent bridging gateway 136, 150 to 154
 - customizing, summary of
 - features/platforms supported 134
 - description 147
 - DIAGnostics parameter 133
 - DiscoverRoutes command 144
 - end system source routing
 - aging out entries 147
 - description 155
 - route discovery 142, 144
 - static routes 145
 - token access priority 147
 - explorer frames, restricting the propagation 142
 - features/platforms supported 134
 - frame size 139
 - GatewayControl parameter 137
 - GatewayVRing parameter 137
 - HoldTime parameter 147
 - LargestFrameSize parameter 139
 - logical ring 139
 - MaxAreRDLimit parameter 142
 - MaxSteRDLimit parameter 142
 - MinAccessPrior parameter 147
 - Mode parameter 140
 - over
 - Frame Relay 1013, 1180
 - SMDS 1051
 - X.25 1100
 - parallel bridges 141
 - passive bridging 139
 - per-port
 - route discovery 142
 - source route bridging 135, 147
 - source route transparent bridging 135, 148
 - source route transparent bridging gateway 136, 148
 - platforms supported 134
 - redundancy 141
 - route designator 150
 - route discovery
 - All Routes Explorer frame 154
 - for end system 155
 - Spanning Tree Explorer frame 155
 - ROUte parameter 145
 - RouteDiscovery parameter 143, 144
 - routing information indicator 149
 - routing table 156
 - security 142
 - spanning tree effects 141
 - SrcRouBridge parameter 135
 - statistics, displaying 132, 1402
 - token ring end station support 134
- troubleshooting 133
- source route transparent bridging 148
- source route transparent bridging gateway
 - connecting SR and TB domains 136
 - TB domain virtual ring
 - number 137
 - token ring frame conversion format 137
 - description 150
 - frame
 - Ethernet 802.2 to/from token ring 802.2 conversion 152
 - maximum size 154
 - GatewayControl parameter 137
 - GatewayVRing parameter 137
 - packet handling
 - SR-to-TB domain 151
 - TB-to-SR domain 152
 - spanning tree loop detection 150
- source route transparent bridging gateway (SRTG), LLC-based token ring to/from Ethernet II conversion 153
- source route transparent gateway. *See* SRTG
- source routing for end systems 155
- Spanning Tree Algorithm. *See* bridging
- Spanning Tree policy, configuring over PPP 122
- specifically routed frame. *See* SRF
- split horizon
 - AppleTalk 635
 - DECnet 646
 - IPX 601
 - RIP-IP 204
 - RIPNG 317
 - VINES 691
 - XNS 702
- spoofing
 - definition 611
 - NCP 591
 - Novell NetWare packets 589 to 593
 - SPX1 593
- SR Service statistics 1402
- SrcRouBridge parameter 135
- SrcSecurity parameter 102
- SRF 154
- SRTG 136
- SscpLinkSta parameter 787
- SSCP-PU session support. *See* NetView
- Service Point
- StartupNET parameter 633, 1084
- StartupNODE parameter 633
- StartupNODe parameter 1084
- static path
 - definition 1006
- static paths 64
- static routes. *See* routes
- static routing tables, maximum entries allowed 1411
- StaticPolicy parameter 200
- statistics display
 - AppleTalk Service 1359
 - ARP Service 1363
 - ATUN Service 1365
 - BGP Service 1366
 - BRidge Service 1367
 - BSC Service 1368
 - CLNP Service 1368, 1369
 - DECnet Service 1370
 - DLSw Service 1371

- DVMRP Service 1373
 - FR Service 1374
 - IDP Service 1374
 - IPX Service 1381
 - ISIS Service 1382
 - LLC2 Service 1384
 - MIP Service 1386
 - MOSPF Service 1386
 - NLSP Service 1387
 - NRIP Service 1389
 - OSPF Service 1390
 - PATH Service 1391
 - PORT Service 1393
 - PPP Service 1394
 - RIPIP Service 1396
 - RIPXNS Service 1397
 - SAP Service 1398
 - SMDS Service 1400
 - SNMP Service 1401
 - SR Service 1402
 - STP Service 1403
 - SYS Service 1404
 - TCP Service 1404
 - UDP Service 1405
 - UDPHelp Service 1406
 - VIP Service 1406
 - X25 Service 1409
 - STATistics parameter 618, 685, 696
 - statistics, data compression 1271
 - status code 1413, 1417
 - STE 155
 - StopBits parameter 807
 - STP Service statistics 1403
 - strings, generating. *See* regular expressions
 - subnet
 - addressing 1330
 - masks 1332
 - variable length with RIPIP 166
 - Subscriber Network Interface 1067
 - SuperStack II
 - configuring ports and paths
 - for local and wide area interfaces 71, 72
 - numbering ports and paths 67
 - virtual ports
 - configuring for wide area interfaces 84
 - definition 77
 - inherited attributes 83
 - models supported on 78
 - over Frame Relay, ATM DXI, and X.25 80
 - over PPP 82
 - over SMDS 82
 - SVC 1211
 - comparing with PVC 1010
 - configuration example 1011
 - configuring 1010
 - definition 1010
 - Switched-56 line
 - configuring 982
 - definition 965
 - Switched Multimegabit Data Service. *See* SMDS
 - switched virtual circuit, X.25 1107
 - switched virtual circuit. *See* SVC
 - switched virtual circuit. *See* SVC
 - switching. *See* local and global switching
 - Synchronous Data Link Control. *See* SDLC
 - synchronous data link control. *See* SDLC
 - SYS Service statistics 1404
 - syslog messages 1417
 - system caller ID. *See* SCID
 - System IP
 - advertising to neighbors 244, ?? to 245
 - ARP packets 245
 - ARP table entry 244
 - configuring 243
 - deleting 244
 - IP addresses allowed 243
 - NAT/firewall issues 245
 - route table entry 243
 - routing issues 244
 - System IP defined 243
-
- T**
 - T1 channelized leased lines 74
 - T1 lines
 - configuring 982
 - definition 965
 - T3 lines
 - definition 965
 - T3 Plus interoperability 1437
 - TACACS database service, mapping to UDP ports 388
 - TCP port mapping, NAT Service 320
 - TCP port numbers 327
 - TCP Service statistics 1404
 - TCP/IP connections, incoming
 - checking network resources 1167
 - domain name service 1142
 - enabling 1159
 - session management. *See* sessions
 - technical support
 - 3Com Knowledgebase Web Services 1495
 - 3Com URL 1495
 - bulletin board service 1496
 - fax service 1496
 - network suppliers 1496
 - product repair 1498
 - telephone lines
 - configuring 978
 - definition 965
 - Telnet
 - access for network management 1296
 - connections
 - incoming 1133
 - outgoing 1117
 - session management. *See* sessions to TCP/IP resources 1161
 - restricting access by address 1298
 - using System IP 243
 - Telnet command 1161
 - TERM Service, X.3 parameter
 - equivalence 1435
 - terminal adapter, using with bandwidth management 966
 - TFTP server 882, 886, 890
 - TFTP service, mapping to UDP ports 388
 - TG parameter 519
 - THreshold parameter 341
 - time service, mapping to UDP ports 387
 - TlmerAck parameter 705
 - TlmerInact parameter 705
 - TlmerReply parameter 705
 - timers, RDP Service 378
 - tinygram compression
 - enabling 1269
 - when to use 1273
 - token ring bridging and IBM connectivity 138
 - token ring I/O module 1437
 - TraceRoute command 163
 - TrafficShaper parameter 1189
 - translation bridging
 - between
 - Ethernet and token ring networks 114
 - configuring 108
 - description 114
 - protocol support 108
 - restrictions
 - for AppleTalk 116
 - for IPX 117
 - translation failure actions, NAT Service 321
 - transmission groups (TGs)
 - adding 551
 - configuring parallel TGs 505
 - deleting 552
 - displaying information for 519
 - TG row configuration for class of service 551
 - TransmitWindow parameter 706
 - transparent bridging
 - description 111
 - over
 - ATM DXI 1215
 - Frame Relay 1012
 - SMDS 1049
 - over MLN 94, 101
 - per port 101
 - setting up 93
 - traps
 - sending in response to events 1287
 - types of audit trail notification 1295
 - troubleshooting
 - AppleTalk router 618
 - APPN router 497
 - Boundary Routing 834
 - bridge 99
 - DCE loopback testing 1319
 - displays
 - active connections 520
 - current adjacent link station status 521
 - current adjacent node status 521
 - current status of Intermediate Session Routing 521
 - failed connections 1166
 - FDDI 1217
 - incoming connections 1137
 - IP multicasting 338
 - IPX router 572
 - OSI router 662
 - outgoing connections 1129
 - RDP Service 380
 - source route bridge 133
 - VINES router 686
 - WAN Extender
 - NETBuilder II troubleshooting commands 957
 - WAN Extender troubleshooting commands 954
 - XNS router 696

tunnel connections
 configuring
 central site bridge/router for
 incoming requests 742
 local switching port groups 751
 terminal end 731, 734
 description 761
 disabling 744
 multicast, configuring 343
 packets
 assigning priority 1280
 encapsulation 761
 terminology 1116
 traffic, prioritizing 745 to 749
 tunnel configuration, verifying 719
 tunnels
 disabling 744
 tunnel endpoints
 using System IP 243
 TUNnelPriority parameter 749, 1280,
 1281

U

UDP Broadcast Helper
 boot request packets 393
 booting clients in order 394
 BOOTP Protocol 396
 BOOTP traffic, relaying 391
 BootpMaxHops parameter 393
 BootpThreshold parameter 393
 broadcast packets, forwarding 395
 configuration
 checking 393
 prerequisites for 388
 statistics, displaying 393
 configuring 388
 for auto startup 884
 for BOOTP 391
 maximum hops for booting 393
 relay BOOTP and DHCP
 traffic 391
 description 387, 395
 DHCP Protocol 396
 DHCP traffic, relaying 391
 mapping service names to UDP
 ports 387
 UDP port mapping, NAT Service 320
 UDP port numbers 327
 UDP Service statistics 1405
 UDPhelp Service statistics 1406
 UME 1211
 UNDefine command 1141
 UNI 1186, 1211
 unnumbered links, running 159, 308
 UpdateTime parameter
 DVMRP 349
 NRIP 577
 RIP 200
 RIPNG 315
 RIPXNS 700
 SAP 577
 VINES 691
 Upgrade Utilities 886
 URL 1495
 User Datagram Protocol (UDP) Broadcast
 Helper. *See* UDP Broadcast Helper
 user-to-network interface. *See* UNI

V

VCC 1211
 VCI 1211
 VINES routing
 client/server support 692
 configuration, verifying 685 to 686
 configuring
 over Frame Relay 1026
 over LANs 683
 over SMDS 1064
 over X.25 1095
 neighbors, assigning symbolic names
 to 687
 network address, router-assigned 683
 network reachability 691
 router 687
 routes 691
 SampleTime parameter 685
 split horizon 691
 STATistics parameter 685
 update packets, transmission
 interval 691
 UpdateTime parameter 691
 VINES Neighbor Table 690
 VINES Routing Table 688
 WAN configurations 684
 VIP Service statistics 1406
 VirBrNum parameter 1300
 VirRingNum parameter 1300
 virtual bridges, configuring for LAN Net
 Manager support 1300
 virtual channel connection. *See* VCC
 virtual channel identifier. *See* VCI
 virtual circuit identifier. *See* VPI.VCI
 virtual path identifier. *See* VPI
 virtual path identifier. *See* VPI.VCI
 virtual paths
 creating for WAN Extender 960
 for WAN Extender
 definition 83
 for WAN Extender leased lines 961
 WAN Extender
 for DS0 dial-up path pool 961
 for HO dial-up path pool 961
 MultiLink Protocol to bind multiple
 paths to single port 83
 setting HO dial-up path pool 961
 setting number of paths 961
 virtual pipe
 allocating bandwidth 966
 definition 1006
 description 1004
 illustrated 1005
 WAN Extender virtual path in 966
 virtual port
 definition 1006
 virtual ports
 configuring for
 APPN over Frame Relay 1018
 disaster recovery over Frame
 Relay 1039
 wide area interfaces 84
 definition 77
 inherited attributes 83
 lack of connectivity 1037
 number supported per platform 78
 over
 ATM 81

Frame Relay, ATM DXI, and
 X.25 80
 PPP 82
 SMDS 82
 platforms supported on 78
 virtual rings, configuring for LAN Net
 Manager support 1300
 VLeasedLine parameter
 L2Tunnel Service 293
 VPI 1211
 VPI.VCI, converting to Frame Relay
 DLCI 1214
 VRRP for Token Ring
 functional address mode 376
 VTp command 1164
 VTP connections, outgoing 1122

W

WAN
 Boundary Routing. *See* Boundary
 Routing
 bridging
 source route over Frame
 Relay 1013, 1180
 source route over SMDS 1051
 source route over X.25 1100
 transparent over ATM 1179
 transparent over Frame
 Relay 1012
 transparent over SMDS 1049
 transparent over X.25 1081,
 1098
 HSS port utilization percentage 1437
 PPP 972
 routing
 AppleTalk over Frame
 Relay 1014 to 1016
 AppleTalk over
 SMDS 1052 to 1056
 AppleTalk over
 X.25 1082 to 1085
 DECnet over SMDS 1056
 DECnet over X.25 1085
 IP over ATM 1181
 IP over Frame Relay 1020
 IP over SMDS 1057
 IP over X.25 1086
 IPX over ATM 1184
 IPX over Frame Relay 1023
 IPX over SMDS 1060
 OSI over Frame Relay 1025
 OSI over SMDS 1062
 OSI over X.25 1092
 VINES over SMDS 1064
 VINES over X.25 1095
 XNS over Frame Relay 1027,
 1216
 XNS over SMDS 1065
 XNS over X.25 1096
 serial ports 1438
 V.35 HSS module placement 1437
 WAN Extender
 Baud parameter
 PATH Service 946
 call filtering 945
 channel bundling 946
 Clock parameter 946
 COMPResType parameter 949

configuration customization 945
 CONfiguration parameter 946, 949
 configuring 931
 CONNect parameter 947
 CONTrol parameter 947
 DialCONTrol parameter 947
 DialNoList parameter 949
 DialPool parameter 948
 DialStatus parameter 950
 DLTest command 946
 ExDevType parameter 948
 how it operates 962
 interconnecting remote LANs with
 central site 931
 interconnection ISDN BRI to ISDN PRI
 configuration example 937
 configuring the NETBuilder II
 procedure 939
 configuring the WAN Extender
 procedure 938
 ISDN H0 Support 945
 leased DS0s to channelized T1
 configuration example 932
 configuring the NETBuilder II 934
 configuring the WAN
 Extender 933
 LineType parameter 948
 model types described 960
 MultiLink Protocol
 multiple paths bound to single
 port 83
 NETBuilder II troubleshooting
 commands 957
 WAN Extender Service
 parameters 957
 OWNer parameter 950
 PathPreference parameter 950
 PATHs parameter 950
 remote connection
 considerations 943
 dial-up options 943
 remote site identification
 options 944
 sample configuration displays 950
 statistics 1408
 statistics, -SYS STATistics
 -WANExtender 1408
 switched 56 circuits configuration
 example 942
 switched 56 circuits. *See*
 Interconnecting ISDN BRI to ISDN PRI
 configuration example
 topology that requires virtual ports 78
 troubleshooting commands
 Caution 954, 957
 setting up WAN Extender
 console 954
 what they do 954
 troubleshooting configurations 953
 virtual paths
 creating 960
 definition 83
 for DS0 dial-up path pool 961
 for H0 dial-up path pool 961
 for leased lines 961
 number NETBuilder II
 supports 931
 setting number of paths 961
 VirtualPort parameter 950
 WAN Extender Manager 962

WAN Extender and NETBuilder II
 configuration 931
 hardware and software
 requirements 932
 WAN setup information 1437
 WanRoutes parameter, SR Service 133
 wide area bridges, configuring 122
 wide area network setup
 information 1437
 World Wide Web (WWW) 1495

X

X.25

AppleTalk routing
 in AppleTalk and non-AppleTalk
 configurations 1082, 1084
 over traffic prioritization 1083
 basic routing 1081
 bridging over prerequisites for 1098
 configuration
 checking 1081
 example 1081
 prerequisites for 1074
 configuring
 AppleTalk routing 1082
 DECnet routing 1085
 for Boundary Routing 828
 IP routing 1086
 OSI routing 1092
 source route bridging 1100
 transparent bridging 1098
 VINES routing 1095
 XNS routing 1096
 your bridge/router 1074
 DECnet routing over traffic
 prioritization 1086
 facilities 1105
 IP routing over traffic
 prioritization 1089
 IPX routing over traffic
 prioritization 1091
 Neighbors parameter 1093
 OSI routing over traffic
 prioritization 1094
 PrefixRoute parameter 1093
 prioritizing traffic
 AppleTalk 1083
 DECnet 1086
 example 1079
 IP 1089
 IPX 1091
 OSI 1094
 procedure 1099
 VINES 1095
 XNS 1097
 profiles, configuration
 parameters 1077
 public or private data network
 (PDN) 1073
 PVC prerequisites for 1102
 source route bridging over 1100,
 1101
 StartupNET parameter 1084
 StartupNODE parameter 1084
 statistics display 1409
 topologies
 fully meshed 1103
 nonmeshed 1103
 partially meshed 1104, 1105
 using virtual ports 1104
 transparent bridging over 94, 1099
 verifying the configuration 1075
 VIP routing over traffic
 prioritization 1095
 XNS routing over traffic
 prioritization 1097
 X.25 configuration options 1271
 X.25 connection service
 incoming. *See* incoming connections
 outgoing. *See* outgoing connections
 X.25 incoming calls, forwarding. *See* local
 and global switching
 X.25 prefix mapping. *See* local and global
 switching
 X.25 profiles 1075, 1076
 X.3 parameters 1435
 X.3-to-TERM Service parameter
 equivalence 1435
 X.500 directory service for incoming OSI
 connections 1145
 X25 Service statistics 1409
 Xerox Network Systems routing. *See* XNS
 routing
 XNS routing
 configuration 694, 696
 configuring over
 ATM DXI 1215
 Frame Relay 1027
 LANs 693
 PPP 693
 SMDS 694, 1065
 X.25 1096
 CONTROL parameter
 IDP 699
 RIPXNS 699
 description 701
 LAN and WAN configuration
 example 697
 network reachability 702
 packets
 error checking, enabling 701
 RIPXNS parameters for
 updates 698
 ROUTE parameter 698
 routes
 dynamic 698
 learning 701
 selection of 702
 static, adding 698
 routing table
 deleting dynamic and static
 routes 702
 displaying 701
 displaying static routes 698
 SampleTime parameter 696
 split horizon
 example 703
 with poison reverse 703
 statistics display
 IDP Service 1374
 RIPXNS Service 1397
 STATistics parameter 696
 UpdateTime parameter 700
 WAN configurations 694
 XSwitch Service. *See* local and global
 switching

Z

- Zone Advertisement Filtering 626
- ZONe parameter 615, 616
- ZoneNetMapping parameter 617, 630

3Com Corporation LIMITED WARRANTY

HARDWARE

3Com warrants its hardware products to be free from defects in workmanship and materials, under normal use and service, for the following lengths of time from the date of purchase from 3Com or its authorized reseller:

| | |
|--|----------|
| Network Interface Cards | Lifetime |
| Other hardware products
unless otherwise specified above | 1 year |
| Spare parts and spares kits | 90 days |

If a product does not operate as warranted above during the applicable warranty period, 3Com shall, at its option and expense, repair the defective product or part, deliver to Customer an equivalent product or part to replace the defective item, or refund to Customer the purchase price paid for the defective product. All products that are replaced will become the property of 3Com. Replacement products may be new or reconditioned. Any replaced or repaired product or part has a ninety (90) day warranty or the remainder of the initial warranty period, whichever is longer.

SOFTWARE

3Com warrants that the software programs licensed from it will perform in substantial conformance to the program specifications therefor for a period of ninety (90) days from the date of purchase from 3Com or its authorized reseller. 3Com warrants the media containing software against failure during the warranty period. No updates are provided. 3Com's sole obligation with respect to this express warranty shall be (at 3Com's discretion) to refund the purchase price paid by Customer for any defective software products, or to replace any defective media with software which substantially conforms to applicable 3Com published specifications. Customer assumes responsibility for the selection of the appropriate applications program and associated reference materials. 3Com makes no warranty or representation that its software products will meet Customer's requirements or work in combination with any hardware or applications software products provided by third parties, that the operation of the software products will be uninterrupted or error free, or that all defects in the software products will be corrected. For any third party products listed in the 3Com software product documentation or specifications as being compatible, 3Com will make reasonable efforts to provide compatibility, except where the non-compatibility is caused by a "bug" or defect in the third party's product.

YEAR 2000 WARRANTY

In addition to the Hardware Products Warranty and Software Products Warranty identified above, 3Com warrants that all Heritage 3Com products sold or licensed to Customer on and after January 1, 1998 that are date sensitive will continue performing properly with regard to such date data on and after January 1, 2000, provided that all other products used by Customer in connection or combination with the 3Com products, including hardware, software, and firmware, accurately exchange date data with the 3Com products, with the exception of those products identified at 3Com's Web site, <http://www.3com.com/products/yr2000.html>, as not meeting this standard. A product is considered a "Heritage 3Com product" if it is a member of a product family which was manufactured by 3Com prior to its merger with US Robotics Corporation. This Year 2000 limited warranty does not apply to Heritage US Robotics Corporation products. If it appears that any such product does not perform properly with regard to such date data on and after January 1, 2000, and Customer notifies 3Com before the later of April 1, 2000, or ninety (90) days after purchase of the product from 3Com or its authorized reseller, 3Com shall, at its option and expense, provide a software update which would effect the proper performance of such product, repair such product, deliver to Customer an equivalent product to replace such product, or if none of the foregoing is feasible, refund to Customer the purchase price paid for such product.

Any software update or replaced or repaired product will carry a Year 2000 Warranty for ninety (90) days or until April 1, 2000, whichever is later.

OBTAINING WARRANTY SERVICE

Customer must contact 3Com's Corporate Service Center or an Authorized 3Com Service Center within the applicable warranty period to obtain warranty service authorization. Dated proof of purchase may be required. Products returned to 3Com's Corporate Service Center must be pre-authorized by 3Com with a Return Material Authorization (RMA) number marked on the outside of the package, and sent prepaid and packaged appropriately for safe shipment, and it is recommended that they be insured. The repaired or replaced item will be shipped to Customer, at 3Com's expense, not later than thirty (30) days after receipt of the defective product by 3Com.

Dead- or Defective-on-Arrival. In the event a product completely fails to function or exhibits a defect in materials or workmanship within the first forty-eight (48) hours of installation but no later than thirty (30) days after the date of purchase, and this is verified by 3Com, it will be considered dead- or defective-on-arrival (DOA) and a replacement shall be provided by advance replacement. The replacement product will normally be shipped not later than three (3) business days after 3Com's verification of the DOA product, but may be delayed due to export or import procedures. When an advance replacement is provided and Customer fails to return the defective product to 3Com within fifteen (15) days after shipment of the replacement, 3Com will charge Customer for the replacement product, at list price.

3Com shall not be responsible for any software, firmware, information, or memory data of Customer contained in, stored on, or integrated with any products returned to 3Com for repair, whether under warranty or not.

WARRANTIES EXCLUSIVE

IF A 3COM PRODUCT DOES NOT OPERATE AS WARRANTED ABOVE, CUSTOMER'S SOLE REMEDY FOR BREACH OF THAT WARRANTY SHALL BE REPAIR, REPLACEMENT, OR REFUND OF THE PURCHASE PRICE PAID, AT 3COM'S OPTION. TO THE FULL EXTENT ALLOWED BY LAW, THE FOREGOING WARRANTIES AND REMEDIES ARE EXCLUSIVE AND ARE IN LIEU OF ALL OTHER WARRANTIES, TERMS, OR CONDITIONS, EXPRESS OR IMPLIED, EITHER IN FACT OR BY OPERATION OF LAW, STATUTORY OR OTHERWISE, INCLUDING WARRANTIES, TERMS, OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND SATISFACTORY QUALITY. 3COM NEITHER ASSUMES NOR AUTHORIZES ANY OTHER PERSON TO ASSUME FOR IT ANY OTHER LIABILITY IN CONNECTION WITH THE SALE, INSTALLATION, MAINTENANCE OR USE OF ITS PRODUCTS.

3COM SHALL NOT BE LIABLE UNDER THIS WARRANTY IF ITS TESTING AND EXAMINATION DISCLOSE THAT THE ALLEGED DEFECT IN THE PRODUCT DOES NOT EXIST OR WAS CAUSED BY CUSTOMER'S OR ANY THIRD PERSON'S MISUSE, NEGLIGENCE, IMPROPER INSTALLATION OR TESTING, UNAUTHORIZED ATTEMPTS TO REPAIR OR MODIFY, OR ANY OTHER CAUSE BEYOND THE RANGE OF THE INTENDED USE, OR BY ACCIDENT, FIRE, LIGHTNING, OR OTHER HAZARD.

LIMITATION OF LIABILITY

TO THE FULL EXTENT ALLOWED BY LAW, 3COM ALSO EXCLUDES FOR ITSELF AND ITS SUPPLIERS ANY LIABILITY, WHETHER BASED IN CONTRACT OR TORT (INCLUDING NEGLIGENCE), FOR INCIDENTAL, CONSEQUENTIAL, INDIRECT, SPECIAL, OR PUNITIVE DAMAGES OF ANY KIND, OR FOR LOSS OF REVENUE OR PROFITS, LOSS OF BUSINESS, LOSS OF INFORMATION OR DATA, OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH THE SALE, INSTALLATION, MAINTENANCE,

USE, PERFORMANCE, FAILURE, OR INTERRUPTION OF ITS PRODUCTS, EVEN IF 3COM OR ITS AUTHORIZED RESELLER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND LIMITS ITS LIABILITY TO REPAIR, REPLACEMENT, OR REFUND OF THE PURCHASE PRICE PAID, AT 3COM'S OPTION. THIS DISCLAIMER OF LIABILITY FOR DAMAGES WILL NOT BE AFFECTED IF ANY REMEDY PROVIDED HEREIN SHALL FAIL OF ITS ESSENTIAL PURPOSE.

DISCLAIMER

Some countries, states, or provinces do not allow the exclusion or limitation of implied warranties or the limitation of incidental or consequential damages for certain products supplied to consumers or the limitation of liability for personal injury, so the above limitations and exclusions may be limited in their application to you. When the implied warranties are not allowed to be excluded in their entirety, they will be limited to the duration of the applicable written warranty. This warranty gives you specific legal rights which may vary depending on local law.

GOVERNING LAW

This Limited Warranty shall be governed by the laws of the State of California, U.S.A. excluding its conflicts of laws principles and excluding the United Nations Convention on Contracts for the International Sale of Goods.

3Com Corporation, 5400 Bayfront Plaza, Santa Clara, CA 95052-8145 (408) 764-5000

[warranty4.doc, DM:3/05/98](#)