

HP 5820X & 5800 Switch Series Layer 3 - IP Services

Configuration Guide

Abstract

This document describes the software features for the HP A Series products and guides you through the software configuration procedures. These configuration guides also provide configuration examples to help you apply software features to different network scenarios.

This documentation is intended for network planners, field technical support and servicing engineers, and network administrators working with the HP A Series products.

Part number: 5998-1632
Software version: Release 1211
Document version: 6W101-20121123



Legal and notice information

© Copyright 2012 Hewlett-Packard Development Company, L.P.

No part of this documentation may be reproduced or transmitted in any form or by any means without prior written consent of Hewlett-Packard Development Company, L.P.

The information contained herein is subject to change without notice.

HEWLETT-PACKARD COMPANY MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Contents

| | |
|--|----|
| ARP configuration | 1 |
| ARP function | 1 |
| ARP message format | 1 |
| Address resolution process | 2 |
| ARP table | 3 |
| Configuring ARP | 4 |
| Configuring a static ARP entry | 4 |
| Configuring the maximum number of dynamic ARP entries for an interface | 4 |
| Setting the age timer for dynamic ARP entries | 5 |
| Enabling dynamic ARP entry check | 5 |
| Configuring ARP quick update | 6 |
| Configuring multicast ARP for NLB | 6 |
| Displaying and maintaining ARP | 7 |
| ARP configuration example | 7 |
| Multicast ARP entry configuration example | 9 |
| Gratuitous ARP configuration | 11 |
| Configuring gratuitous ARP | 12 |
| Proxy ARP configuration | 13 |
| Proxy ARP | 13 |
| Local proxy ARP | 14 |
| Enabling proxy ARP | 14 |
| Displaying and maintaining proxy ARP | 15 |
| Proxy ARP configuration example | 15 |
| Local proxy ARP configuration example in case of port isolation | 16 |
| Local proxy ARP configuration example in super VLAN | 17 |
| Local proxy ARP configuration example in isolate-user-VLAN | 18 |
| ARP snooping configuration | 20 |
| Operation | 20 |
| Configuring ARP snooping | 20 |
| Displaying and maintaining ARP snooping | 20 |
| IP addressing configuration | 21 |
| Address classes | 21 |
| Special IP addresses | 22 |
| Subnetting and masking | 22 |
| Configuring IP addresses | 23 |
| Assigning an IP address to an interface | 23 |
| IP addressing configuration example | 23 |
| Configuring IP unnumbered | 25 |
| Configuration prerequisites | 25 |
| Configuration procedure | 25 |
| Displaying and maintaining IP addressing | 26 |
| DHCP overview | 27 |
| DHCP address allocation | 27 |
| Allocation mechanisms | 27 |
| Dynamic IP address allocation process | 28 |
| IP address lease extension | 28 |
| Message format | 29 |

| | |
|---|-----------|
| DHCP options | 30 |
| Self-defined options | 31 |
| Protocols and standards | 34 |
| DHCP server configuration | 35 |
| Application environment | 35 |
| DHCP address pool | 35 |
| Address pool types | 35 |
| Common address pool structure | 35 |
| Address pool selection principles | 36 |
| IP address allocation sequence | 36 |
| Configuration task list | 37 |
| Configuring a DHCP server address pool | 37 |
| Creating a DHCP address pool | 38 |
| Configuring an address allocation mode for a common address pool | 38 |
| Configuring dynamic address allocation for an extended address pool | 40 |
| Configuring a domain name suffix for the client | 40 |
| Configuring DNS servers for the client | 41 |
| Configuring WINS servers and NetBIOS node type for the client | 41 |
| Configuring BIMS server information for the client | 42 |
| Configuring gateways for the client | 43 |
| Configuring Option 184 parameters for the client with voice service | 43 |
| Configuring the TFTP server and Bootfile name for the client | 43 |
| Specifying a server's IP address for the client | 44 |
| Configuring self-defined DHCP options | 45 |
| Enabling DHCP | 46 |
| Enabling the DHCP server on an interface | 46 |
| Applying an extended address pool on an interface | 47 |
| Configuring the DHCP server security functions | 47 |
| Configuration prerequisites | 47 |
| Enabling unauthorized DHCP server detection | 47 |
| Configuring IP address conflict detection | 48 |
| Enabling Option 82 handling | 48 |
| Specifying the threshold for sending trap messages | 49 |
| Configuration prerequisites | 49 |
| Configuration procedure | 49 |
| Displaying and maintaining the DHCP server | 49 |
| DHCP server configuration examples | 50 |
| Static IP address assignment configuration example | 50 |
| Dynamic IP address assignment configuration example | 51 |
| Self-defined option configuration example | 53 |
| Troubleshooting DHCP server configuration | 54 |
| DHCP relay agent configuration | 55 |
| Application environment | 55 |
| Fundamentals | 55 |
| DHCP relay agent support for Option 82 | 56 |
| Configuration task list | 57 |
| Enabling DHCP | 57 |
| Enabling the DHCP relay agent on an interface | 58 |
| Correlating a DHCP server group with a relay agent interface | 58 |
| Configuring the DHCP relay agent security functions | 59 |
| Creating static bindings and enabling address check | 59 |
| Configuring periodic refresh of dynamic client entries | 59 |
| Enabling unauthorized DHCP server detection | 60 |
| Enabling DHCP starvation attack protection | 60 |

| | |
|---|-----------|
| Enabling offline detection..... | 61 |
| Configuring the DHCP relay agent to release an IP address..... | 62 |
| Configuring the DHCP relay agent to support Option 82..... | 62 |
| Displaying and maintaining the DHCP relay agent..... | 64 |
| DHCP relay agent configuration examples..... | 64 |
| DHCP relay agent Option 82 support configuration example..... | 65 |
| Troubleshooting DHCP relay agent configuration..... | 66 |
| DHCP client configuration..... | 67 |
| Enabling the DHCP client on an interface..... | 67 |
| Displaying and maintaining the DHCP client..... | 67 |
| DHCP client configuration example..... | 67 |
| DHCP snooping configuration..... | 70 |
| Snooping functions..... | 70 |
| Ensuring DHCP clients to obtain IP addresses from authorized DHCP servers..... | 70 |
| Recording DHCP client IP-to-MAC mappings..... | 70 |
| Application environment of trusted ports..... | 71 |
| DHCP snooping support for Option 82..... | 72 |
| Configuration task list..... | 73 |
| Configuring DHCP snooping basic functions..... | 74 |
| Configuring DHCP snooping to support Option 82..... | 75 |
| Configuring DHCP snooping entries backup..... | 76 |
| Enabling DHCP starvation attack protection..... | 77 |
| Enabling DHCP-REQUEST message attack protection..... | 77 |
| Configuring DHCP packet rate limit..... | 78 |
| Displaying and maintaining DHCP snooping..... | 78 |
| Configuration examples..... | 79 |
| DHCP snooping configuration example..... | 79 |
| DHCP snooping Option 82 support configuration example..... | 80 |
| BOOTP client configuration..... | 81 |
| Application..... | 81 |
| Dynamically obtaining an IP address..... | 81 |
| Protocols and standards..... | 81 |
| Configuring an interface to dynamically obtain an IP address through BOOTP..... | 82 |
| Displaying and maintaining BOOTP client configuration..... | 82 |
| BOOTP client configuration example..... | 82 |
| IPv4 DNS configuration..... | 83 |
| Static domain name resolution..... | 83 |
| Dynamic domain name resolution..... | 83 |
| DNS proxy..... | 84 |
| DNS spoofing..... | 85 |
| Configuring the IPv4 DNS client..... | 86 |
| Configuring static domain name resolution..... | 86 |
| Configuring dynamic domain name resolution..... | 86 |
| Configuring the DNS proxy..... | 88 |
| Configuring DNS spoofing..... | 88 |
| Configuration prerequisites..... | 88 |
| Configuration procedure..... | 88 |
| Displaying and maintaining IPv4 DNS..... | 89 |
| IPv4 DNS configuration examples..... | 89 |
| Static domain name resolution configuration example..... | 89 |
| Dynamic domain name resolution configuration example..... | 90 |
| DNS proxy configuration example..... | 93 |
| Troubleshooting IPv4 DNS configuration..... | 94 |

| | |
|--|-----|
| IPv6 DNS configuration | 95 |
| Configuring the IPv6 DNS client | 95 |
| Configuring static domain name resolution | 95 |
| Configuring dynamic domain name resolution | 95 |
| Displaying and maintaining IPv6 DNS | 96 |
| IPv6 DNS configuration examples | 96 |
| Static domain name resolution configuration example | 96 |
| Dynamic domain name resolution configuration example | 97 |
| IP performance optimization configuration | 103 |
| Enabling reception and forwarding of directed broadcasts to a directly connected network | 103 |
| Enabling reception of directed broadcasts to a directly connected network | 103 |
| Enabling forwarding of directed broadcasts to a directly connected network | 104 |
| Configuration example | 104 |
| Configuring TCP attributes | 105 |
| Configuring the TCP send/receive buffer size | 105 |
| Configuring TCP timers | 105 |
| Configuring ICMP to send error packets | 106 |
| Configuration procedure | 107 |
| Enabling ICMP extension support | 107 |
| Configuration procedure | 108 |
| Displaying and maintaining IP performance optimization | 109 |
| IRDP configuration | 110 |
| Working mechanism | 110 |
| Terminology | 111 |
| Protocols and standards | 111 |
| Configuring IRDP | 112 |
| IRDP configuration example | 113 |
| UDP Helper configuration | 115 |
| Configuring UDP Helper | 115 |
| Displaying and maintaining UDP Helper | 116 |
| Configuration examples | 116 |
| UDP Helper configuration example | 116 |
| IPv6 basics configuration | 117 |
| IPv6 features | 117 |
| IPv6 addresses | 118 |
| IPv6 NDP | 121 |
| IPv6 PMTU discovery | 124 |
| IPv6 transition technologies | 124 |
| Protocols and standards | 125 |
| Configuration task list | 125 |
| Configuring basic IPv6 functions | 126 |
| Enabling IPv6 | 126 |
| Configuring an IPv6 global unicast address | 126 |
| Configuring an IPv6 link-local address | 128 |
| Configure an IPv6 anycast address | 130 |
| Configuring IPv6 NDP | 130 |
| Configuring a static neighbor entry entry | 130 |
| Configuring the maximum number of neighbors dynamically learned | 131 |
| Configuring RA message-related parameters | 131 |
| Configuring the maximum number of attempts to send an NS message for DAD | 134 |
| Setting the age timer for ND entries | 134 |
| Configuring ND snooping | 135 |

| | |
|---|------------|
| Enabling ND proxy | 136 |
| Configuring PMTU discovery | 138 |
| Configuring a static PMTU for a specified IPv6 address | 138 |
| Configuring the aging time for dynamic PMTUs | 138 |
| Configuring IPv6 TCP properties | 139 |
| Configuring ICMPv6 packet sending | 139 |
| Configuring the maximum ICMPv6 error packets sent in an interval | 139 |
| Enabling replying to multicast echo requests | 140 |
| Enabling sending of ICMPv6 time exceeded messages | 140 |
| Enabling sending of ICMPv6 destination unreachable messages | 141 |
| Displaying and maintaining IPv6 basics configuration | 141 |
| IPv6 configuration example | 143 |
| Troubleshooting IPv6 basics configuration | 148 |
| DHCPv6 overview | 149 |
| Address/prefix assignment | 149 |
| Rapid assignment involving two messages | 149 |
| Assignment involving four messages | 150 |
| Address/prefix lease renewal | 150 |
| Stateless DHCPv6 configuration | 151 |
| Operation | 151 |
| Protocols and standards | 152 |
| DHCPv6 server configuration | 153 |
| Application environment | 153 |
| Basic concepts | 153 |
| Prefix selection process | 154 |
| Configuration task list | 154 |
| Configuration prerequisites | 155 |
| Enabling the DHCPv6 server | 155 |
| Creating a prefix pool | 155 |
| Configuring a DHCPv6 address pool | 155 |
| Applying the address pool to an interface | 156 |
| Displaying and maintaining the DHCPv6 server | 157 |
| DHCPv6 server configuration example | 157 |
| DHCPv6 relay agent configuration | 161 |
| Application environment | 161 |
| Operation | 161 |
| Configuring the DHCPv6 relay agent | 162 |
| Configuration prerequisites | 162 |
| Configuration procedure | 162 |
| Displaying and maintaining the DHCPv6 relay agent | 163 |
| DHCPv6 relay agent configuration example | 163 |
| DHCPv6 client configuration | 165 |
| Configuration prerequisites | 165 |
| Configuration procedure | 165 |
| Displaying and maintaining the DHCPv6 client | 165 |
| Stateless DHCPv6 configuration example | 166 |
| DHCPv6 snooping configuration | 168 |
| Overview | 168 |
| Ensuring DHCPv6 clients to obtain IPv6 addresses from authorized DHCPv6 servers | 168 |
| Recording IP-to-MAC mappings of DHCPv6 clients | 169 |
| Enabling DHCPv6 snooping | 169 |
| Configuring a DHCPv6 snooping trusted port | 169 |

| | |
|--|------------|
| Configuring the maximum number of DHCPv6 Snooping entries an interface can learn | 170 |
| Displaying and maintaining DHCPv6 snooping | 170 |
| DHCPv6 snooping configuration example | 170 |
| Tunneling configuration | 172 |
| IPv4/IPv6 tunnels | 172 |
| IPv6 over IPv4 tunnel | 173 |
| IPv4 over IPv4 tunnel | 175 |
| IPv4/IPv6 over IPv6 tunnel | 176 |
| GRE tunnel | 177 |
| Protocols and standards | 178 |
| Configuration task list | 179 |
| Configuring a tunnel interface | 179 |
| Configuration prerequisites | 179 |
| Configuration procedure | 179 |
| Configuring an IPv6 manual tunnel | 180 |
| Configuration prerequisites | 180 |
| Configuration procedure | 181 |
| Configuration example | 182 |
| Configuring a 6to4 tunnel | 185 |
| Configuration prerequisites | 185 |
| Configuration procedure | 186 |
| 6to4 tunnel configuration example | 187 |
| Configuring an ISATAP tunnel | 189 |
| Configuration prerequisites | 189 |
| Configuration procedure | 190 |
| Configuration example | 191 |
| Configuring an IPv4 over IPv4 tunnel | 194 |
| Configuration prerequisites | 194 |
| Configuration procedure | 194 |
| Configuration example | 195 |
| Configuring an IPv4 over IPv6 tunnel | 198 |
| Configuration prerequisites | 198 |
| Configuration procedure | 198 |
| Configuration example | 199 |
| Configuring an IPv6 over IPv6 tunnel | 203 |
| Configuration prerequisites | 203 |
| Configuration procedure | 203 |
| Configuration example | 204 |
| Configuring a GRE over IPv4 tunnel | 207 |
| Configuration prerequisites | 207 |
| Configuration procedure | 208 |
| Configuration example | 209 |
| Configuring a GRE over IPv6 tunnel | 212 |
| Configuration prerequisites | 212 |
| Configuration procedure | 212 |
| Configuration example | 213 |
| Displaying and maintaining tunneling configuration | 217 |
| Troubleshooting tunneling configuration | 218 |
| Support and other resources | 219 |
| Contacting HP | 219 |
| Subscription service | 219 |
| Related information | 219 |
| Documents | 219 |
| Websites | 219 |

| | |
|-------------------|-----|
| Conventions | 220 |
| Index | 222 |

ARP configuration

The Layer 3 Ethernet interface refers to the Ethernet port that can perform IP routing and inter-VLAN routing. set an Ethernet port as a Layer 3 Ethernet interface by using **port link-mode route** (see the *Layer 2—LAN Switching Configuration Guide*).

ARP function

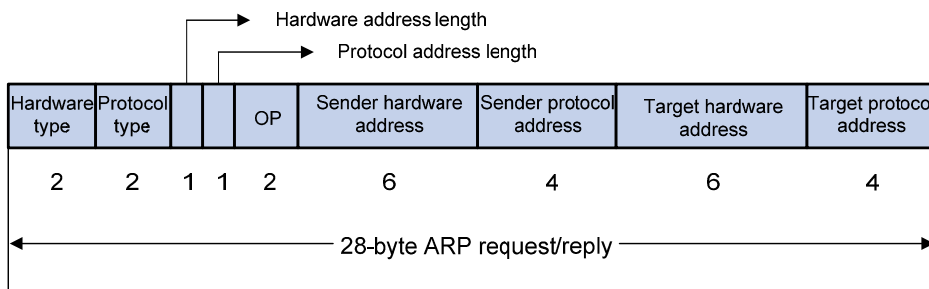
The ARP is used to resolve an IP address into a physical address (Ethernet MAC address, for example).

In an Ethernet LAN, a switch uses ARP to resolve the IP address of the next hop to the corresponding MAC address.

ARP message format

ARP messages are classified into ARP requests and ARP replies. Figure 1 shows the format of the ARP request and reply. Numbers in the figure refer to field lengths.

Figure 1 ARP message format



The following describe the fields in Figure 1.

- **Hardware type**—The hardware address type. The value 1 represents Ethernet.
- **Protocol type**—The type of the protocol address to be mapped. The hexadecimal value 0x0800 represents IP.
- **Hardware address length and protocol address length**—Length, in bytes, of a hardware address and protocol address. For an Ethernet address, the value of the hardware address length field is 6. For an IPv4 address, the value of the protocol address length field is 4.
- **OP**—Operation code. The type of the ARP message. The value 1 represents an ARP request and 2 represents an ARP reply.
- **Sender hardware address**—Hardware address of the switch sending the message.
- **Sender protocol address**—Protocol address of the switch sending the message.
- **Target hardware address**—Hardware address of the switch the message is being sent to.
- **Target protocol address**—Protocol address of the switch the message is being sent to.

Address resolution process

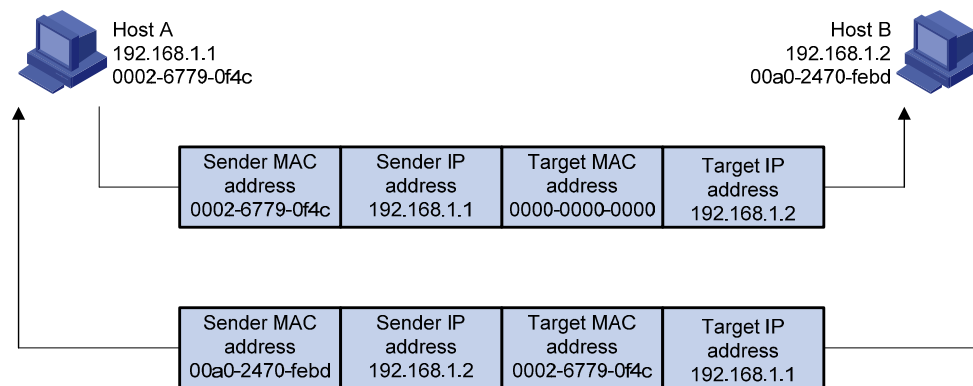
If Host A and Host B are on the same subnet and Host A sends a packet to Host B, as shown in Figure 2:

1. Host A looks in its ARP table to see whether there is an ARP entry for Host B. If yes, Host A uses the MAC address in the entry to encapsulate the IP packet into a data link layer frame and sends the frame to Host B.
2. If Host A finds no entry for Host B, Host A buffers the packet and broadcasts an ARP request using the following information:
 - Source IP address and source MAC address: Host A's own IP address and the MAC address
 - Target IP address: Host B's IP address
 - Target MAC address: An all-zero MAC address.

Because the ARP request is broadcast, all hosts on this subnet can receive the request, but only the requested host (Host B) processes the request.

3. Host B compares its own IP address with the target IP address in the ARP request. If they are the same, Host B:
 - Adds the sender IP address and sender MAC address to its ARP table
 - Encapsulates its MAC address into an ARP reply
 - Unicasts the ARP reply to Host A.
4. After receiving the ARP reply, Host A:
 - Adds the MAC address of Host B to its ARP table
 - Encapsulates the MAC address in the IP packet and sends it to Host B.

Figure 2 ARP address resolution process



If Host A and Host B are not on the same subnet:

1. Host A sends an ARP request to the gateway. The target IP address in the ARP request is the IP address of the gateway.
2. After obtaining the MAC address of the gateway from an ARP reply, Host A sends the packet to the gateway.
3. If the gateway maintains the ARP entry of Host B, it forwards the packet to Host B directly; if not, it broadcasts an ARP request, in which the target IP address is the IP address of Host B.
4. After obtaining the MAC address of Host B, the gateway sends the packet to Host B.

ARP table

After obtaining a host's MAC address, the switch adds the IP-to-MAC mapping to its own ARP table. This mapping is used for forwarding packets with the same destination in future.

An ARP table contains either category of ARP entries: dynamic or static.

Dynamic ARP entry

A dynamic entry is automatically created and maintained by ARP. It can age out, be updated by a new ARP packet, or be overwritten by a static ARP entry. A dynamic ARP entry is removed when its age timer expires or the interface goes down.

Static ARP entry

A static ARP entry is manually configured and maintained. It does not age out or cannot be overwritten by a dynamic ARP entry.

Static ARP entries protect communication between devices, because attack packets cannot modify the IP-to-MAC mapping in a static ARP entry.

Static ARP entries can be long or short.

- A long static ARP entry can be directly used to forward packets directly, because it includes not only the IP address and MAC address, but also a configured VLAN and outbound interface.
- A short static ARP entry includes only an IP address and a MAC address. It cannot be used to forward data directly. When a short static ARP entry matches an IP packet to be forwarded, the switch sends an ARP request first. If the sender IP and MAC addresses in the received ARP reply are the same as those in the short static ARP entry, the switch adds the interface receiving the ARP reply to the short static ARP entry. Then the entry can be used for forwarding IP packets.

Usually ARP dynamically resolves IP addresses to MAC addresses, without manual intervention.

To allow communication with a switch using a fixed IP-to-MAC mapping, configure a short static ARP entry for it. To allow communication with a device through a specific interface in a specific VLAN and using a fixed IP-to-MAC mapping, configure a long static ARP entry for it.

Configuring ARP

Configuring a static ARP entry

CAUTION:

- The *vlan-id* argument must be the ID of an existing VLAN which corresponds to the ARP entries. In addition, the Ethernet interface following the argument must belong to that VLAN. A VLAN interface must be created for the VLAN.
- The IP address of the VLAN interface corresponding to the *vlan-id* argument must belong to the same subnet as the IP address specified by the *ip-address* argument.

A static ARP entry is effective when the device it corresponds to works normally. However, when a VLAN or VLAN interface is deleted, any static ARP entry corresponding to it is also deleted (if it is a long static ARP entry) or becomes unresolved (if it is a short and resolved static ARP entry).

To configure a static ARP entry:

| Step | Command | Remarks |
|--|--|--|
| 1. Enter system view. | system-view | — |
| 2. Configure a long static ARP entry. | arp static ip-address mac-address vlan-id interface-type interface-number [vpn-instance vpn-instance-name] | Required. No long static ARP entry is configured by default. |
| 3. Configure a short static ARP entry. | arp static ip-address mac-address [vpn-instance vpn-instance-name] | Required. No short static ARP entry is configured by default. |

Configuring the maximum number of dynamic ARP entries for an interface

To set the maximum number of dynamic ARP entries that an interface can learn:

| Step | Command | Remarks |
|---|--|---|
| 1. Enter system view. | system-view | — |
| 2. Enter Ethernet interface view. | interface interface-type interface-number | — |
| 3. Set the maximum number of dynamic ARP entries that an interface can learn. | arp max-learning-number number | Optional. 16,384 by default for HP 5800 switch series 8192 by default for HP 5820X switch series If the value of the number argument is set to 0, the interface is disabled from learning dynamic ARP entries. |

Setting the age timer for dynamic ARP entries

Each dynamic ARP entry in the ARP table has a limited lifetime, called age timer. The age timer of a dynamic ARP entry is reset each time the dynamic ARP entry is used. Dynamic ARP entries that are not used before expiration are deleted from the ARP table. adjust the age timer for dynamic ARP entries according to the actual network condition.

To set the age timer for dynamic ARP entries:

| Step | Command | Remarks |
|---|--|-------------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Set the age timer for dynamic ARP entries. | arp timer aging <i>aging-time</i> | Optional. 20 minutes by default. |

Enabling dynamic ARP entry check

The dynamic ARP entry checks function controls whether the switch supports dynamic ARP entries with multicast MAC addresses.

When dynamic ARP entry check is enabled, the switch cannot learn dynamic ARP entries containing multicast MAC addresses.

When dynamic ARP entry check is disabled, the switch can learn dynamic ARP entries containing multicast MAC addresses.

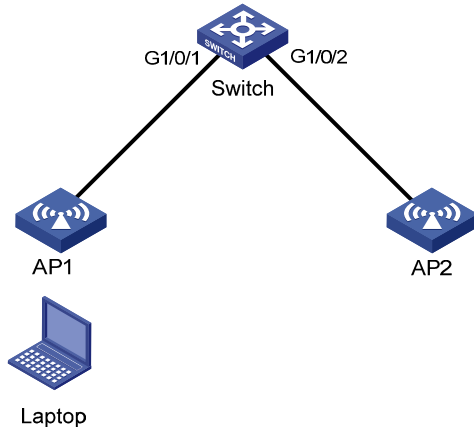
To enable ARP entry check:

| Step | Command | Remarks |
|------------------------------------|-------------------------|----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable dynamic ARP entry check. | arp check enable | Optional. Enabled by default. |

Configuring ARP quick update

As shown in Figure 3, the laptop frequently roams between AP 1 and AP 2. This affects the mapping between its MAC address and outbound interface on the switch. If the switch does not update its ARP table immediately after the outbound interface changes, it can fail to communicate with the laptop.

Figure 3 ARP quick update application scenario



With ARP quick update enabled, the switch updates the corresponding ARP entry immediately after the change of the mapping between a MAC address and an outbound interface to ensure nonstop data forwarding.

To enable ARP quick update:

| Step | Command | Remarks |
|----------------------------|--|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable ARP quick update | mac-address station-move quick-notify enable | Required. Disabled by default. |

HP recommends enabling ARP quick update only in WLANs.

Configuring multicast ARP for NLB

Microsoft NLB is developed on Windows Server for server clustering.

NLB supports load sharing and redundancy among servers within a cluster. It enables fast service switchover when a server fails. To work with NLB, the switch must forward network traffic to all or specified servers in the cluster. Each server filters out unexpected traffic. In a medium or small data center using the Windows Server operating system, the cooperation between the switch and NLB is very important.

To enable the switch to forward network traffic to all or specified servers, Microsoft NLB provides the following forwarding modes:

- **Unicast mode**—NLB assigns each NLB node a common MAC address, which is the cluster MAC address. Each node changes the source MAC address of each sent packet so that the switch cannot add the cluster MAC address to its MAC table. Because the cluster MAC address is unknown to the switch, packets destined to it are forwarded on all ports of the switch.
- **Multicast mode**—NLB uses a multicast MAC address (virtual MAC address), such as 0300-5e11-1111, for network communication.

- **IGMP multicast mode**—The switch sends packets only to the ports that connect to NLB nodes rather than all ports.

Multicast ARP only applies to multicast-mode NLB. For more information about NLB, see the related documents of Windows Server.

To configure multicast ARP entries:

| Step | Command | Remarks |
|--|--|-----------|
| 1. Disable the ARP entry check function. | undo arp check enable | Required. |
| 2. Configure a static ARP entry. | arp static ip-address mac-address vlan-id interface-type interface-number [vpn-instance vpn-instance-name] | Optional. |
| 3. Configure a static multicast MAC address entry. | mac-address multicast mac-address interface interface-list vlan vlan-id | Required. |

For more information about **mac-address multicast**, see *IP Multicast Command Reference*.

Displaying and maintaining ARP

| Task | Command | Remarks |
|---|---|------------------------|
| Display ARP entries in the ARP table. | display arp [[all dynamic static] [slot slot-number] vlan vlan-id interface interface-type interface-number] [count verbose] [[{ begin exclude include } regular-expression] | Available in any view |
| Display the ARP entry for a specified IP address. | display arp ip-address [slot slot-number] [verbose] [[{ begin exclude include } regular-expression] | Available in any view |
| Display the ARP entries for a specified VPN instance. | display arp vpn-instance vpn-instance-name [count] [[{ begin exclude include } regular-expression] | Available in any view |
| Display the age timer for dynamic ARP entries. | display arp timer aging [[{ begin exclude include } regular-expression] | Available in any view |
| Clear ARP entries from the ARP table. | reset arp { all dynamic static slot slot-number interface interface-type interface-number } | Available in user view |

Clearing ARP entries from the ARP table can cause communication failures.

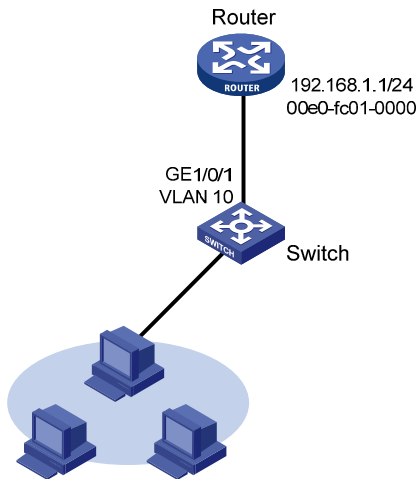
ARP configuration example

Network requirements

As shown in [Figure 4](#), hosts are connected to Switch, which is connected to Router through interface GigabitEthernet 1/0/1 belonging to VLAN 10. The IP address of Router is 192.168.1.1/24. The MAC address of Router is 00e0-fc01-0000.

To prevent malicious users from attacking the switch and enhance security for communications between the router and the switch, configure a static ARP entry for the router on the switch.

Figure 4 Network diagram for configuring static ARP entries



Configuration procedure

Create VLAN 10.

```
<Switch> system-view
[Switch] vlan 10
[Switch-vlan10] quit
```

Add interface GigabitEthernet 1/0/1 to VLAN 10.

```
[Switch] interface GigabitEthernet 1/0/1
[Switch-GigabitEthernet1/0/1] port link-type trunk
[Switch-GigabitEthernet1/0/1] port trunk permit vlan 10
[Switch-GigabitEthernet1/0/1] quit
```

Create interface VLAN-interface 10 and configure its IP address.

```
[Switch] interface vlan-interface 10
[Switch-vlan-interface10] ip address 192.168.1.2 24
[Switch-vlan-interface10] quit
```

Configure a static ARP entry with IP address 192.168.1.1 and MAC address 00e0-fc01-0000. The outgoing interface corresponding to the static ARP entry is GigabitEthernet1/0/1 belonging to VLAN 10.

```
[Switch] arp static 192.168.1.1 00e0-fc01-0000 10 GigabitEthernet 1/0/1
```

View information about static ARP entries.

```
[Switch] display arp static
```

| IP Address | MAC Address | VLAN ID | Interface | Ageing Type |
|-------------|----------------|---------|-----------|-------------|
| 192.168.1.1 | 00e0-fc01-0000 | 10 | GE1/0/1 | N/A S |

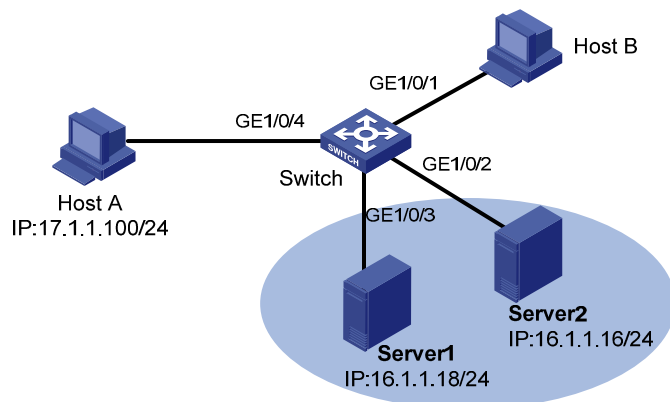
Multicast ARP entry configuration example

Network requirements

As shown in Figure 5, multicast support is enabled for the Microsoft NLB service in a small-sized data center. To enable the switch to cooperate with NLB, configure the switch as follows:

- Disable the ARP entry check function so that the switch can learn dynamic ARP entries containing multicast MAC addresses.
- Configure a static multicast ARP entry.
- Configure a static multicast MAC address entry so that only interfaces GigabitEthernet 1/0/2 and GigabitEthernet 1/0/3 can receive multicast information.

Figure 5 Diagram for multicast ARP entry configuration



- Interfaces GigabitEthernet 1/0/1, GigabitEthernet 1/0/2, and GigabitEthernet 1/0/3 belong to VLAN 1, and the IP address of VLAN-interface 1 is 16.1.1.30/24.
- Interface GigabitEthernet 1/0/4 belongs to VLAN 2, and the IP address of the VLAN-interface 2 is 17.1.1.1/24.
- The gateway IP address of Host A is 17.1.1.1/24.
- The gateway IP address of Server 1 and Server 2 is 16.1.1.30/24.

Configuration procedure

This example only gives the configurations on the switch. For the NLB configuration on the servers, see related documents for the Windows server.

This example is only applicable for NLB multicast mode, and assumes that the virtual IP address of the servers is 16.1.1.100/24, and the virtual MAC address is 03bf-1001-0164.

Assign an IP address to VLAN-interface 2.

```
<Switch> system-view
[Switch] vlan 2
[Switch-vlan2] port GigabitEthernet 1/0/4
[Switch-vlan2] quit
[Switch] interface vlan-interface 2
[Switch-Vlan-interface2] ip address 17.1.1.1 255.255.255.0
[Switch-Vlan-interface2] quit
```

Assign an IP address to VLAN-interface 1.

```
[Switch] interface vlan-interface 1
[Switch-Vlan-interface1] ip address 16.1.1.30 255.255.255.0
```

```
[Switch-Vlan-interface1] quit
#Disable the ARP entry check function.
[Switch] undo arp check enable
# Configure a static multicast ARP entry.
[Switch] arp static 16.1.1.100 03bf-1001-0164 1 GigabitEthernet 1/0/2
# Configure a static multicast MAC address entry.
[Switch] mac-address multicast 03bf-1001-0164 interface GigabitEthernet 1/0/2
GigabitEthernet 1/0/3 vlan 1
```

Verification

Ping the virtual IP address of the servers from the switch.

```
[Switch] ping 16.1.1.100
  PING 16.1.1.100: 56 data bytes, press CTRL_C to break
    Reply from 16.1.1.100: bytes=56 Sequence=1 ttl=128 time=4 ms
    Reply from 16.1.1.100: bytes=56 Sequence=2 ttl=128 time=2 ms
    Reply from 16.1.1.100: bytes=56 Sequence=3 ttl=128 time=1 ms
    Reply from 16.1.1.100: bytes=56 Sequence=4 ttl=128 time=2 ms
    Reply from 16.1.1.100: bytes=56 Sequence=5 ttl=128 time=12 ms
--- 16.1.1.100 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/4/12 ms
```

Gratuitous ARP configuration

In a gratuitous ARP packet, the sender IP address and the target IP address are both the IP address of the switch issuing the packet, the sender MAC address is the MAC address of the switch, and the target MAC address is the broadcast address ff:ff:ff:ff:ff:ff.

A switch sends gratuitous ARP packets for the following purposes:

- To determine whether its IP address is already used by another device. If the IP address is already used, the device issuing the gratuitous ARP packet is informed by an ARP reply of the conflict.
- To inform other devices about the change of its MAC address so that they can update their ARP entries.

Enabling learning of gratuitous ARP packets

With this feature enabled, a switch receiving a gratuitous ARP packet adds the sender IP and MAC addresses carried in the packet to its ARP table if no corresponding ARP entry exists. If a corresponding ARP entry is found, the switch updates the ARP entry.

After this feature is disabled, the switch uses the address information in the received gratuitous ARP packets to update the existing ARP entries only, but not to create new ARP entries.

Configuring periodic sending of gratuitous ARP packets

Enabling a switch to periodically send gratuitous ARP packets helps downstream devices update their corresponding ARP entries or MAC entries in time. This feature can be used to prevent gateway spoofing, prevent ARP entries from aging out, and prevent the virtual IP address of a VRRP group from being used by a host.

- Prevent gateway spoofing

When an attacker sends forged gratuitous ARP packets to the hosts on a network, the traffic destined for the gateway from the hosts is sent to the attacker instead. As a result, the hosts cannot access the external network.

To prevent such gateway spoofing attacks, enable the gateway to send gratuitous ARP packets containing its primary IP address and manually configured secondary IP addresses at a specific interval. In this way, each host can learn correct gateway address information.

- Prevent ARP entries from aging out

If network traffic is heavy or a host's CPU usage is high, received ARP packets can be discarded or not processed in time. Eventually, the dynamic ARP entries on the receiving host ages out, and the traffic between the host and the corresponding devices are interrupted until the host re-creates the ARP entries.

To prevent this problem, enable the gateway to send gratuitous ARP packets periodically. The gratuitous ARP packets contain the gateway's primary IP address or one of its manually configured secondary IP addresses. In this way, the receiving host can update ARP entries in time and thus ensure traffic continuity.

- Prevent the virtual IP address of a VRRP group from being used by a host

The master router of a VRRP group can periodically send gratuitous ARP packets to the hosts on the local network, so that the hosts can update local ARP entries and avoid using the virtual IP address of the VRRP group.

If the virtual IP address of the VRRP group is associated with a virtual MAC address, the sender MAC address in the gratuitous ARP packet takes the virtual MAC address of the virtual router. If the virtual IP address of the VRRP group is associated with the real MAC address of an interface, the sender MAC

address in the gratuitous ARP packet takes the MAC address of the interface on the master router in the VRRP group.

For more information about VRRP, see *High Availability Configuration Guide*.

Configuring gratuitous ARP

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enable learning of gratuitous ARP packets. | gratuitous-arp-learning enable | Optional. Enabled by default. |
| 3. Enable the switch to send gratuitous ARP packets upon receiving ARP requests from another subnet. | gratuitous-arp-sending enable | Required. By default, the switch does not send gratuitous ARP packets upon receiving ARP requests from another subnet. |
| 4. Enter interface view. | interface <i>interface-type interface-number</i> | — |
| 5. Enable periodic sending of gratuitous ARP packets and set the sending interval. | arp send-gratuitous-arp [interval <i>milliseconds</i>] | Required. Disabled by default. |

Enable periodic sending of gratuitous ARP packets on a maximum of 1024 interfaces.

Periodic sending of gratuitous ARP packets only takes effect when the link of the enabled interface goes up and an IP address has been assigned to the interface.

If you change the interval for sending gratuitous ARP packets, the configuration is effective at the next sending interval.

The frequency of sending gratuitous ARP packets can be much lower than is expected if this function is enabled on multiple interfaces, if each interface is configured with multiple secondary IP addresses, or if a small sending interval is configured in such cases.

Proxy ARP configuration

Proxy ARP includes common proxy ARP and local proxy ARP.

- Common proxy ARP allows communication when a sending host considers the receiving host to be on the same subnet, but the receiving host actually resides on a different subnet.
- Local proxy ARP allows communication between hosts that reside on the same subnet but are isolated at Layer 2.

In both cases, a device located between the two hosts must respond to the request with the MAC address of the receiving interface to allow Layer 3 communication between the two hosts. This is achieved by proxy ARP, which hides the physical details of the network.

Proxy ARP involves common proxy ARP and local proxy ARP, which are described in the following sections.

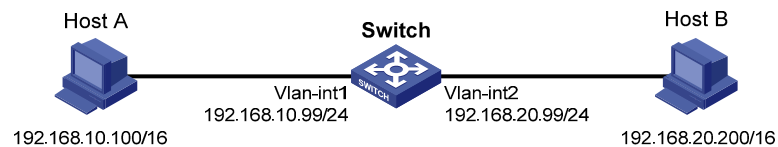
The term proxy ARP in the following sections of this chapter refers to common proxy ARP unless otherwise specified.

Proxy ARP

A proxy ARP enabled device allows hosts that reside on different subnets to communicate.

As shown in [Figure 6](#), Switch connects to two subnets through Vlan-interface1 and Vlan-interface2. The IP addresses of the two interfaces are 192.168.10.99/24 and 192.168.20.99/24. Host A and Host B are assigned the same prefix 192.168.0.0. Host A connects to Vlan-interface1 and Host B connects to Vlan-interface2.

Figure 6 Application environment of proxy ARP



Because Host A considers that Host B is on the same network, it broadcasts an ARP request for the MAC address of Host B. Host B, however, cannot receive this request because it is in a different broadcast domain.

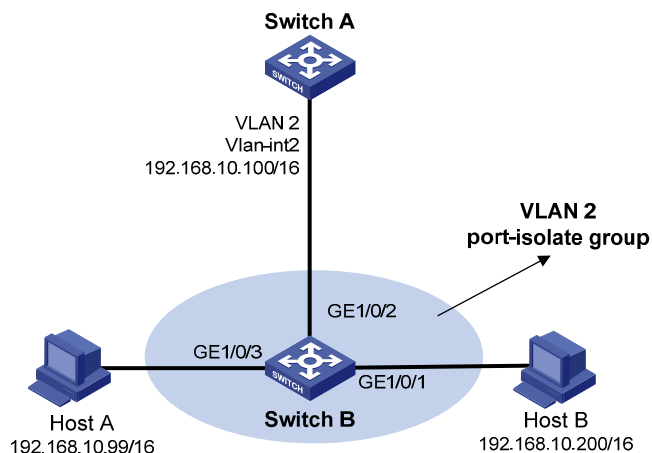
Enable proxy ARP on Vlan-interface1 of Switch so that Switch can reply to the ARP request from Host A with the MAC address of Vlan-interface1, and forward packets sent from Host A to Host B. In this case, Switch acts like a proxy of Host B.

A main advantage of proxy ARP is that enable it on a single router without disturbing routing tables of other switches in the network. Proxy ARP acts as the gateway for hosts that are not configured with a default gateway or do not have routing capability.

Local proxy ARP

As shown in Figure 7, Host A and Host B belong to VLAN 2, but are isolated at Layer 2. Host A connects to GigabitEthernet1/0/3 while Host B connects to GigabitEthernet1/0/1. Enable local proxy ARP on switch A to allow Layer 3 communication between the two hosts.

Figure 7 Application environment of local proxy ARP



In one of the following cases, you need to enable local proxy ARP:

- Hosts connecting to different isolated Layer 2 ports in the same VLAN need to communicate at Layer 3.
- If a super VLAN is configured, hosts in different sub VLANs of the super VLAN need to communicate at Layer 3.
- If an isolate-user-VLAN is configured, hosts in different secondary VLANs of the isolate-user-VLAN need to communicate at Layer 3.

Enabling proxy ARP

Enable proxy ARP in VLAN interface view/Layer 3 Ethernet interface view

| Step | Command | Remarks |
|--------------------------|---|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface <i>interface-type interface-number</i> | — |
| 3. Enable proxy ARP. | proxy-arp enable | Required. Disabled by default. |

Enable local proxy ARP in VLAN interface view/Layer 3 Ethernet interface view

| Step | Command | Remarks |
|----------------------------|---|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface <i>interface-type interface-number</i> | — |
| 3. Enable local proxy ARP. | local-proxy-arp enable [ip-range <i>startIP to endIP</i>] | Required. Disabled by default. |

Displaying and maintaining proxy ARP

| Task | Command | Remarks |
|---|---|-----------------------|
| Display whether proxy ARP is enabled. | display proxy-arp [interface <i>interface-type interface-number</i>] [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display whether local proxy ARP is enabled. | display local-proxy-arp [interface <i>interface-type interface-number</i>] [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |

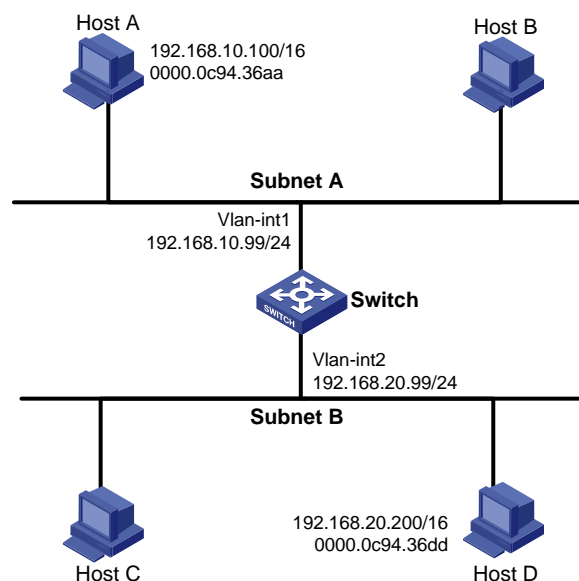
Proxy ARP configuration example

Network requirements

As shown in Figure 8, Host A and Host D have the same prefix and mask (IP addresses of Host A and Host D are 192.168.10.100/16 and 192.168.20.200/16 respectively), but they are located on different subnets separated by the switch. Host A belongs to VLAN 1 and Host D belongs to VLAN 2. As a result, Host D cannot receive or respond to any ARP request from Host A.

To enable communication between the two hosts, configure proxy ARP on the switch.

Figure 8 Network diagram for proxy ARP



Configuration procedure

```
# Create VLAN 2.
<Switch> system-view
[Switch] vlan 2
[Switch-vlan2] quit

# Specify the IP address of interface VLAN-interface 1.
[Switch] interface vlan-interface 1
[Switch-Vlan-interface1] ip address 192.168.10.99 255.255.255.0

# Enable proxy ARP on interface VLAN-interface 1.
[Switch-Vlan-interface1] proxy-arp enable
[Switch-Vlan-interface1] quit

# Specify the IP address of interface VLAN-interface 2.
[Switch] interface vlan-interface 2
[Switch-Vlan-interface2] ip address 192.168.20.99 255.255.255.0

# Enable proxy ARP on interface VLAN-interface 2.
[Switch-Vlan-interface2] proxy-arp enable
```

After completing preceding configurations, use **ping** to verify the connectivity between Host A and Host D.

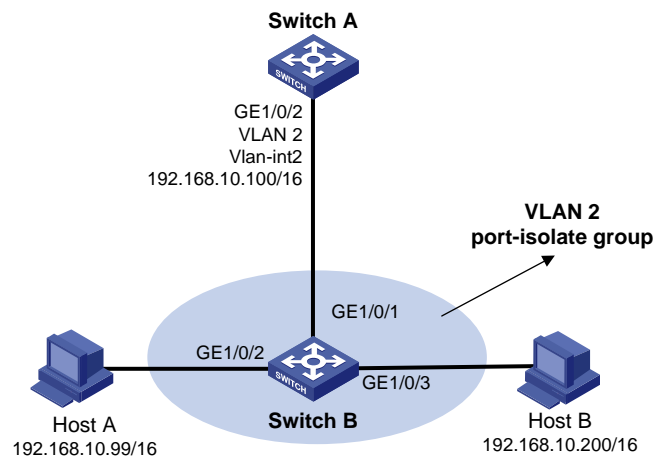
Local proxy ARP configuration example in case of port isolation

Network requirements

As shown in [Figure 9](#), Host A and Host B belong to the same VLAN, and connect to Switch B via GigabitEthernet 1/0/2 and GigabitEthernet 1/0/3 respectively. Switch B connects to Switch A via GigabitEthernet 1/0/1.

Configure port isolation on GigabitEthernet 1/0/3 and GigabitEthernet 1/0/2 of Switch B to isolate Host A from Host B at Layer 2. Enable local proxy ARP on Switch A to allow communication between Host A and Host B at Layer 3.

Figure 9 Network diagram for local proxy ARP between isolated ports



Configuration procedure

1. Configure Switch B

```
# Add GigabitEthernet 1/0/3, GigabitEthernet 1/0/1 and GigabitEthernet 1/0/2 to VLAN 2.
Configure port isolation on Host A and Host B.
```

```
<SwitchB> system-view
```

```

[SwitchB] vlan 2
[SwitchB-vlan2] port gigabitethernet 1/0/1
[SwitchB-vlan2] port gigabitethernet 1/0/2
[SwitchB-vlan2] port gigabitethernet 1/0/3
[SwitchB-vlan2] quit
[SwitchB] interface gigabitethernet 1/0/2
[SwitchB-GigabitEthernet1/0/2] port-isolate enable
[SwitchB-GigabitEthernet1/0/2] quit
[SwitchB] interface gigabitethernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] port-isolate enable
[SwitchB-GigabitEthernet1/0/3] quit

```

2. Configure Switch A

Create VLAN 2, and add GigabitEthernet 1/0/2 to VLAN 2.

```

<SwitchA> system-view
[SwitchA] vlan 2
[SwitchA-vlan2] port GigabitEthernet 1/0/2
[SwitchA-vlan2] quit
[SwitchA] interface vlan-interface 2
[SwitchA-Vlan-interface2] ip address 192.168.10.100 255.255.0.0

```

The ping operation from Host A to Host B is unsuccessful because they are isolated at Layer 2.

Configure local proxy ARP to allow communication between Host A and Host B at Layer 3.

```

[SwitchA-Vlan-interface2] local-proxy-arp enable

```

The ping operation from Host A to Host B is successful after the configuration.

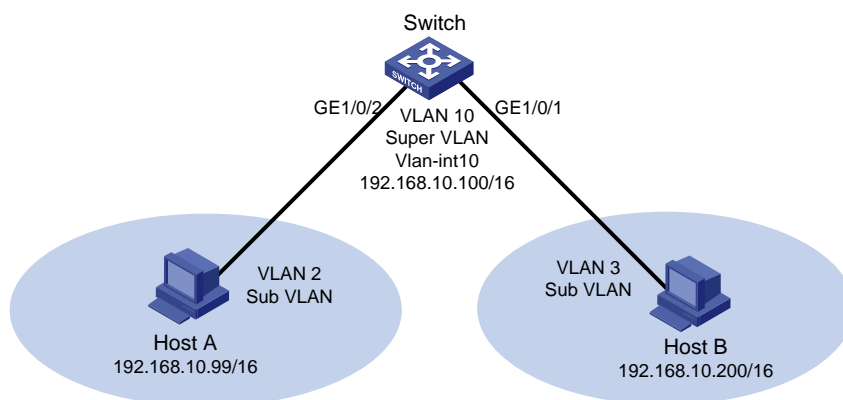
Local proxy ARP configuration example in super VLAN

Network requirements

Figure 10 shows a super VLAN, VLAN 10, with the interface IP address 192.168.10.100/16 and sub-VLANs (VLAN 2 and VLAN 3). GigabitEthernet 1/0/2 belongs to VLAN 2 and GigabitEthernet 1/0/1 belongs to VLAN 3. Host A belongs to VLAN 2 and connects to GigabitEthernet 1/0/2 of the switch. Host B belongs to VLAN 3 and connects to GigabitEthernet 1/0/1 of the switch.

As Host A and Host B belong to different sub-VLANs, they are isolated at Layer 2. Configure local proxy ARP on the switch to allow Layer 3 communication between Host A and Host B.

Figure 10 Network diagram for local proxy ARP in super VLAN



Configuration procedure

Create the super VLAN and the sub-VLANs. Add GigabitEthernet 1/0/2 to VLAN 2 and GigabitEthernet 1/0/1 to VLAN 3. Configure the IP address 192.168.10.100/16 for the interface of VLAN 10.

```
<Switch> system-view
[Switch] vlan 2
[Switch-vlan2] port GigabitEthernet 1/0/2
[Switch-vlan2] quit
[Switch] vlan 3
[Switch-vlan3] port GigabitEthernet 1/0/1
[Switch-vlan3] quit
[Switch] vlan 10
[Switch-vlan10] supervlan
[Switch-vlan10] subvlan 2 3
[Switch-vlan10] quit
[Switch] interface vlan-interface 10
[Switch-Vlan-interface10] ip address 192.168.10.100 255.255.0.0
```

The ping operation from Host A to Host B is unsuccessful because they are isolated at Layer 2.

Configure the local proxy ARP to implement Layer 3 communication between sub-VLANs.

```
[Switch-Vlan-interface10] local-proxy-arp enable
```

The ping operation from Host A to Host B is successful after the configuration.

Local proxy ARP configuration example in isolate-user-VLAN

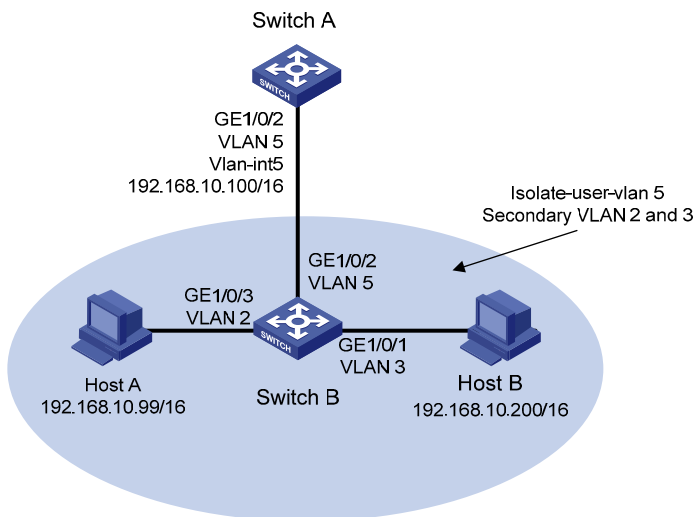
Network requirements

As shown in [Figure 11](#), Switch B is attached to Switch A. VLAN 5 on Switch B is an isolate-user-VLAN, which includes uplink port GigabitEthernet 1/0/2 and two secondary VLANs, VLAN 2 and VLAN 3. GigabitEthernet 1/0/3 belongs to VLAN 2, and GigabitEthernet 1/0/1 belongs to VLAN 3.

Host A belongs to VLAN 2 and connects to GigabitEthernet 1/0/3 of the switch B. Host B belongs to VLAN 3 and connects to GigabitEthernet 1/0/1 of the switch B.

As Host A and Host B belong to different secondary VLANs, they are isolated at Layer 2. Configure local proxy ARP on Switch A to implement Layer 3 communication between Host A and Host B.

Figure 11 Network diagram for local proxy ARP configuration in isolate-user-VLAN



Configuration procedure

1. Configure Switch B

Create VLAN 2, VLAN 3, and VLAN 5 on Switch B. Add GigabitEthernet 1/0/3 to VLAN 2, GigabitEthernet 1/0/1 to VLAN 3, and GigabitEthernet 1/0/2 to VLAN 5. Configure VLAN 5 as the isolate-user-VLAN, and VLAN 2 and VLAN 3 as secondary VLANs. Configure the mappings between isolate-user-VLAN and the secondary VLANs.

```
<SwitchB> system-view
[SwitchB] vlan 2
[SwitchB-vlan2] port GigabitEthernet 1/0/3
[SwitchB-vlan2] quit
[SwitchB] vlan 3
[SwitchB-vlan3] port GigabitEthernet 1/0/1
[SwitchB-vlan3] quit
[SwitchB] vlan 5
[SwitchB-vlan5] port GigabitEthernet 1/0/2
[SwitchB-vlan5] isolate-user-vlan enable
[SwitchB-vlan5] quit
[SwitchB] isolate-user-vlan 5 secondary 2 3
```

2. Configure Switch A

Create VLAN 5 and add GigabitEthernet 1/0/2 to it.

```
<SwitchA> system-view
[SwitchA] vlan 5
[SwitchA-vlan5] port GigabitEthernet 1/0/2
[SwitchA-vlan5] quit
[SwitchA] interface vlan-interface 5
[SwitchA-Vlan-interface5] ip address 192.168.10.100 255.255.0.0
```

The ping operation from Host A to Host B is unsuccessful because they are isolated at Layer 2.

Configure local proxy ARP to implement Layer 3 communication between VLAN 2 and VLAN 3.

```
[SwitchA-Vlan-interface5] local-proxy-arp enable
```

The ping operation from Host A to Host B is successful after the configuration.

ARP snooping configuration

The ARP snooping feature is used in Layer 2 switching networks. It creates ARP snooping entries using ARP packets, and the entries can be used by manual-mode MFF to answer ARP requests from a gateway.

For more information about MFF, see *Security Configuration Guide*.

Operation

If ARP snooping is enabled on a VLAN of a device, ARP packets received by the interfaces of the VLAN are redirected to the CPU. The CPU uses ARP packets to create ARP snooping entries comprising source IP and MAC addresses, VLAN and receiving port information.

The aging time and valid period of an ARP snooping entry are 25 minutes and 15 minutes, respectively. If an ARP snooping entry is not updated within 15 minutes, it becomes invalid and cannot be used. After that, if an ARP packet whose source IP and MAC addresses correspond with the entry is received, the entry becomes valid, and its age timer restarts. If the age timer of an ARP entry expires, the entry is removed.

If the ARP snooping device receives an ARP packet that has the same sender IP address as but a different sender MAC address from a valid ARP snooping entry, it considers that an attack occurs. An ARP snooping entry conflict occurs in this case. As a result, the ARP snooping entry becomes invalid and is removed after 25 minutes.

Configuring ARP snooping

| Step | Command | Remarks |
|-------------------------|----------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter VLAN view. | vlan <i>vlan-id</i> | — |
| 3. Enable ARP snooping. | arp-snooping enable | Required. Disabled by default. |

Displaying and maintaining ARP snooping

| Task | Command | Remarks |
|-------------------------------|---|------------------------|
| Display ARP snooping entries. | display arp-snooping [ip <i>ip-address</i> vlan <i>vlan-id</i>] [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Remove ARP snooping entries. | reset arp-snooping [ip <i>ip-address</i> vlan <i>vlan-id</i>] | Available in user view |

IP addressing configuration

Address classes

IP addressing uses a 32-bit address to identify each host on a network. To make addresses easier to read, they are written in dotted decimal notation, each address being four octets in length. For example, address 00001000000000010000000100000001 in binary is written as 10.1.1.1.

Each IP address breaks down into two parts:

- **Net ID**—Identifies a network. The first several bits of a net ID, known as the class field or class bits, identify the class of the IP address.
- **Host ID**—Identifies a host on a network.

IP addresses are divided into five classes, as shown in Figure 12. The shaded areas represent the address class. The first three classes are widely used.

Figure 12 IP address classes

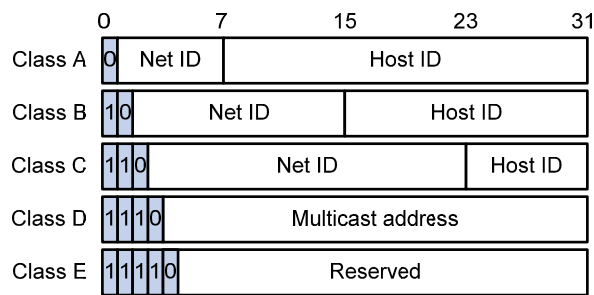


Table 1 describes the address ranges of these five classes.

Table 1 IP address classes and ranges

| Class | Address range | Remarks |
|-------|------------------------------|---|
| A | 0.0.0.0 to 127.255.255.255 | The IP address 0.0.0.0 is used by a host at startup for temporary communication. This address is never a valid destination address. Addresses starting with 127 are reserved for loopback test. Packets destined to these addresses are processed locally as input packets rather than sent to the link. |
| B | 128.0.0.0 to 191.255.255.255 | — |
| C | 192.0.0.0 to 223.255.255.255 | — |
| D | 224.0.0.0 to 239.255.255.255 | Multicast addresses. |
| E | 240.0.0.0 to 255.255.255.255 | Reserved for future use except for the broadcast address 255.255.255.255. |

Special IP addresses

The following IP addresses are for special use, and they cannot be used as host IP addresses.

- **IP address with an all-zero net ID**—Identifies a host on the local network. For example, IP address 0.0.0.16 indicates the host with a host ID of 16 on the local network.
- **IP address with an all-zero host ID**—Identifies a network.
- **IP address with an all-one host ID**—Identifies a directed broadcast address. For example, a packet with the destination address of 192.168.1.255 is broadcast to all hosts on the network 192.168.1.0.

Subnetting and masking

Subnetting divides a network down into smaller networks called subnets by using some bits of the host ID to create a subnet ID.

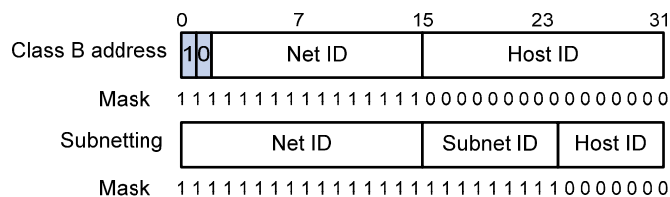
Masking identifies the boundary between the host ID and the combination of net ID and subnet ID.

Each subnet mask comprises 32 bits that correspond to the bits in an IP address. In a subnet mask, the consecutive ones represent the net ID and subnet ID, and consecutive zeros represents the host ID.

Before being subnetted, Class A, B, and C networks use these default masks (also called natural masks): 255.0.0.0, 255.255.0.0, and 255.255.255.0 respectively.

Figure 13 shows how a Class B network is subnetted.

Figure 13 Subnet a Class B network



Subnetting increases the number of addresses that cannot be assigned to hosts. Therefore, using subnets means accommodating somewhat fewer hosts

For example, a Class B network without subnetting can accommodate 1022 more hosts than the same network subnetted into 512 subnets.

- **Without subnetting**—65,534 hosts ($2^{16} - 2$). (The two deducted addresses are the broadcast address, which has an all-one host ID, and the network address, which has an all-zero host ID.)
- **With subnetting**—Using the first 9 bits of the host-id for subnetting provides 512 (2^9) subnets. However, only 7 bits remain available for the host ID. This allows 126 ($2^7 - 2$) hosts in each subnet, a total of 64,512 hosts (512×126).

Configuring IP addresses

An interface must have an IP address to communicate with other hosts. either manually assign an IP address to an interface, or configure the interface to obtain an IP address through BOOTP, or DHCP. If you change the way an interface obtains an IP address, the new IP address overwrites the previous one.

This chapter only covers how to assign an IP address manually. For information about how to obtain an IP address through BOOTP or DHCP, see “[DHCP overview](#)” and “[BOOTP client configuration](#).”

Assigning an IP address to an interface

CAUTION:

- An interface can have only one primary IP address. A newly configured primary IP address overwrites the previous one.
 - You cannot assign secondary IP addresses to an interface that obtains an IP address through BOOTP, or DHCP.
 - The primary and secondary IP addresses you assign to the interface can be located on the same network segment, but different interfaces on your device must reside on different network segments.
-

You can assign an interface multiple IP addresses, one primary and multiple secondaries.

You only need to assign the primary address to an interface. In some cases, you need to assign secondary IP addresses to the interface. For example, if the interface connects to two subnets, to enable the device to communicate with all hosts on the LAN, you need to assign a primary IP address and a secondary IP address to the interface.

To assign an IP address to an interface:

| Step | Command | Remarks |
|---|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Assign an IP address to the interface. | ip address ip-address { mask-length mask } [sub] | Required. No IP address is assigned by default. |

IP addressing configuration example

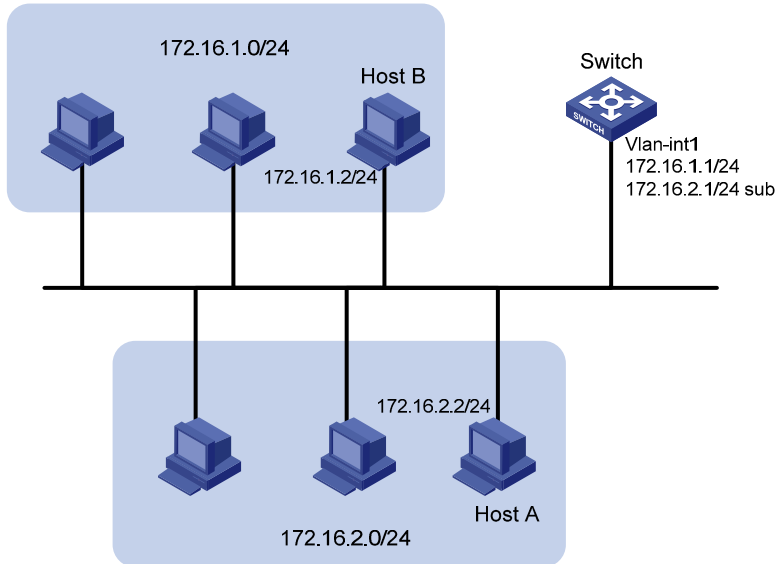
Network requirements

As shown in [Figure 14](#), a port in VLAN 1 on a switch is connected to a LAN comprising two segments: 172.16.1.0/24 and 172.16.2.0/24.

To enable the hosts on the two network segments to communicate with the external network through the switch and to enable the hosts on the LAN can communicate with each other,

- Assign a primary IP address and a secondary IP address to VLAN-interface 1 on the switch.
- Set the primary IP address of the router as the gateway address of the PCs on subnet 172.16.1.0/24, and the secondary IP address of the router as the gateway address of the PCs on subnet 172.16.2.0/24.

Figure 14 Network diagram for IP addressing configuration



Configuration procedure

Assign a primary IP address and a secondary IP address to VLAN-interface 1.

```
<Switch> system-view
```

```
[Switch] interface vlan-interface 1
```

```
[Switch-Vlan-interface1] ip address 172.16.1.1 255.255.255.0
```

```
[Switch-Vlan-interface1] ip address 172.16.2.1 255.255.255.0 sub
```

Set the gateway address to 172.16.1.1 on the PCs attached to subnet 172.16.1.0/24, and to 172.16.2.1 on the PCs attached to subnet 172.16.2.0/24.

Ping a host on subnet 172.16.1.0/24 from the switch to check the connectivity.

```
<Switch> ping 172.16.1.2
```

```
PING 172.16.1.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 172.16.1.2: bytes=56 Sequence=1 ttl=255 time=25 ms
```

```
Reply from 172.16.1.2: bytes=56 Sequence=2 ttl=255 time=27 ms
```

```
Reply from 172.16.1.2: bytes=56 Sequence=3 ttl=255 time=26 ms
```

```
Reply from 172.16.1.2: bytes=56 Sequence=4 ttl=255 time=26 ms
```

```
Reply from 172.16.1.2: bytes=56 Sequence=5 ttl=255 time=26 ms
```

```
--- 172.16.1.2 ping statistics ---
```

```
5 packet(s) transmitted
```

```
5 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 25/26/27 ms
```

The output information shows that the switch can communicate with the hosts on subnet 172.16.1.0/24.

Ping a host on subnet 172.16.2.0/24 from the switch to check the connectivity.

```
<Switch> ping 172.16.2.2
PING 172.16.2.2: 56 data bytes, press CTRL_C to break
  Reply from 172.16.2.2: bytes=56 Sequence=1 ttl=255 time=25 ms
  Reply from 172.16.2.2: bytes=56 Sequence=2 ttl=255 time=26 ms
  Reply from 172.16.2.2: bytes=56 Sequence=3 ttl=255 time=26 ms
  Reply from 172.16.2.2: bytes=56 Sequence=4 ttl=255 time=26 ms
  Reply from 172.16.2.2: bytes=56 Sequence=5 ttl=255 time=26 ms

--- 172.16.2.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 25/25/26 ms
```

The output information shows that the switch can communicate with the hosts on subnet 172.16.2.0/24.

Ping a host on subnet 172.16.1.0/24 from a host on subnet 172.16.2.0/24 to check the connectivity. Host B can be successfully pinged from Host A.

Configuring IP unnumbered

Logically, to enable IP on an interface, you must assign this interface a unique IP address. Yet, you can borrow an IP address already configured on one of other interfaces on your device instead. This interface is called "IP unnumbered" and the interface borrowing the IP address is called "IP unnumbered."

You can use IP unnumbered to save IP addresses either when IP addresses are inadequate or when an interface is brought up only for occasional use.

Configuration prerequisites

Assign a primary IP address to the interface from which you want to borrow the IP address. Alternatively, you can configure the interface to obtain one through BOOTP or DHCP.

Configuration procedure

CAUTION:

- Layer 3 Ethernet interfaces, and loopback interfaces cannot borrow IP addresses of other interfaces, but other interfaces can borrow IP addresses of these interfaces.
 - An interface cannot borrow an IP address from an unnumbered interface.
 - Multiple interfaces can use the same unnumbered IP address.
 - If an interface has multiple IP addresses, only the primary IP address can be borrowed.
 - The IP address of the borrowing interface varies with that of the borrowed interface. If an IP address is configured for the borrowed interface, the IP address of the borrowing interface is the same as that of the borrowed interface; if no IP address is configured for the borrowed interface, no IP address is assigned for the borrowing interface.
-

To configure IP unnumbered on an interface:

| Step | Command | Remarks |
|---|--|---|
| 1. Enter system view. | system-view | — |
| 2. Enter tunnel interface view. | interface tunnel <i>number</i> | — |
| 3. Specify the current interface to borrow the IP address of the specified interface. | ip address unnumbered interface <i>interface-type interface-number</i> | Required. The interface does not borrow IP addresses from other interfaces by default. |

Displaying and maintaining IP addressing

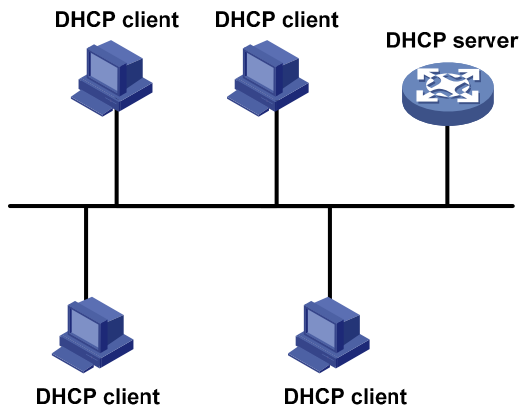
| Task | Command | Remarks |
|---|---|-----------------------|
| Display IP configuration information for a specified Layer 3 interface or all Layer 3 interfaces. | display ip interface [<i>interface-type interface-number</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display brief IP configuration information for a specified Layer 3 interface or all Layer 3 interfaces. | display ip interface [<i>interface-type</i> [<i>interface-number</i>]] brief [{ begin exclude include } <i>regular-expression</i>] | Available in any view |

DHCP overview

The DHCP provides a framework to assign configuration information to network devices. It uses the client/server model.

A typical DHCP application, as shown in [Figure 15](#), includes a DHCP server and multiple clients (PCs and laptops).

Figure 15 A typical DHCP application



A DHCP client can obtain an IP address and other configuration parameters from a DHCP server on another subnet via a DHCP relay agent. For more information about the DHCP relay agent, see [“DHCP relay agent configuration.”](#)

DHCP address allocation

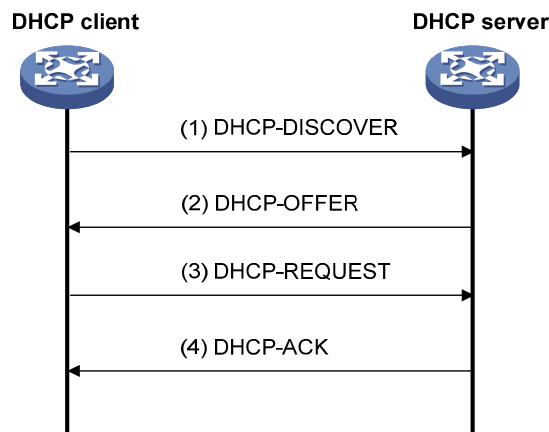
Allocation mechanisms

DHCP supports the following mechanisms for IP address allocation.

- **Static allocation**—The network administrator assigns an IP address to a client like a WWW server, and DHCP conveys the assigned address to the client.
- **Automatic allocation**—DHCP assigns a permanent IP address to a client.
- **Dynamic allocation**—DHCP assigns an IP address to a client for a limited period of time, which is called a lease. Most DHCP clients obtain their addresses in this way.

Dynamic IP address allocation process

Figure 16 Dynamic IP address allocation process



1. The client broadcasts a DHCP-DISCOVER message to locate a DHCP server.
2. A DHCP server offers configuration parameters, such as an IP address to the client in a DHCP-OFFER message. The sending mode of the DHCP-OFFER message is determined by the flag field in the DHCP-DISCOVER message. For related information, see “[Message format.](#)”
3. If several DHCP servers send offers to the client, the client accepts the first received offer, and broadcasts it in a DHCP-REQUEST message to formally request the IP address.
4. All DHCP servers receive the DHCP-REQUEST message, but only the server from which the client accepts the offered IP address returns a DHCP-ACK message to the client, confirming that the IP address has been allocated to the client, or a DHCP-NAK message, denying the IP address allocation.

After the client receives the DHCP-ACK message, it broadcasts a gratuitous ARP packet to verify whether the IP address assigned by the server is already in use. If the client receives no response within the specified time, the client uses the assigned IP address. Otherwise, the client sends a DHCP-DECLINE message to the server and requests an IP address again.

IP addresses offered by other DHCP servers are still assignable to other clients.

IP address lease extension

The IP address dynamically allocated by a DHCP server to a client has a lease. When the lease expires, the IP address is reclaimed by the DHCP server. To continue using the IP address, the client must extend the lease duration.

When half lease duration elapses, the DHCP client sends the DHCP server a DHCP-REQUEST unicast to extend the lease duration. Depending on availability of the IP address, the DHCP server returns a DHCP-ACK unicast confirming that the client’s lease duration has been extended, or a DHCP-NAK unicast denying the request.

If the client receives no reply, it broadcasts another DHCP-REQUEST message for lease extension after 7/8 lease duration elapses.

Message format

Figure 17 shows the DHCP message format, which is based on the BOOTP message format although DHCP uses some of the fields in significantly different ways. The numbers in parentheses indicate the size of each field in bytes.

Figure 17 DHCP message format

| | | | | |
|--------------------|---|-----------|-----------|----------|
| 0 | 7 | 15 | 23 | 31 |
| op (1) | | htype (1) | | hlen (1) |
| hops (1) | | | | |
| xid (4) | | | | |
| secs (2) | | | flags (2) | |
| ciaddr (4) | | | | |
| yiaddr (4) | | | | |
| siaddr (4) | | | | |
| giaddr (4) | | | | |
| chaddr (16) | | | | |
| sname (64) | | | | |
| file (128) | | | | |
| options (variable) | | | | |

- **Op**—Message type defined in option field. 1 = REQUEST, 2 = REPLY
- **Htype, hlen**—Hardware address type and length of a DHCP client.
- **Hops**—Number of relay agents a request message traveled.
- **Xid**—Transaction ID, a random number chosen by the client to identify an IP address allocation.
- **Secs**—Filled in by the client, the number of seconds elapsed since the client began address acquisition or renewal process. This field is reserved and set to 0.
- **Flags**—The leftmost bit is defined as the BROADCAST (B) flag. If this flag is set to 0, the DHCP server sent a reply back by unicast; if this flag is set to 1, the DHCP server sent a reply back by broadcast. The remaining bits of the flags field are reserved for future use.
- **Ciaddr**—Client IP address.
- **Yiaddr**—'your' (client) IP address, assigned by the server.
- **Siaddr**—Server IP address, from which the client obtained configuration parameters.
- **Giaddr**—IP address of the first relay agent a request message traveled.
- **Chaddr**—Client hardware address.
- **Sname**—Server host name, from which the client obtained configuration parameters.
- **File**—Bootfile name and path information, defined by the server to the client.
- **Options**—Optional parameters field that is variable in length, which includes the message type, lease, domain name server IP address, and WINS IP address.

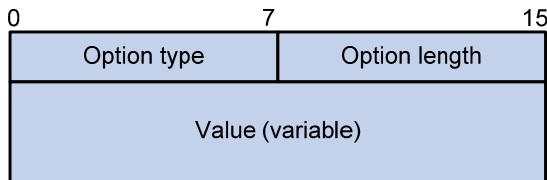
DHCP options

DHCP uses the same message format as BOOTP, but DHCP uses the Option field to carry information for dynamic address allocation and to provide additional configuration information to clients.

DHCP uses the option field in DHCP messages to carry control information and network configuration parameters, implementing dynamic address allocation and providing more network configuration information for clients.

Figure 18 shows the DHCP option format.

Figure 18 DHCP option format



Common DHCP options:

- **Option 3**—Router option. It specifies the gateway address to be assigned to the client.
- **Option 6**—DNS server option. It specifies the DNS server IP address to be assigned to the client.
- **Option 33**—Static route option. It specifies a list of classful static routes (the destination addresses in these static routes are classful) that a client should add to its routing table. If Option 121 exists, Option 33 is ignored.
- **Option 51**—IP address lease option.
- **Option 53**—DHCP message type option. It identifies the type of the DHCP message.
- **Option 55**—Parameter request list option. It is used by a DHCP client to request specified configuration parameters. The option contains values that correspond to the parameters requested by the client.
- **Option 60**—Vendor class identifier option. It is used by a DHCP client to identify its vendor, and by a DHCP server to distinguish DHCP clients by vendor class and assign specific IP addresses for the DHCP clients.
- **Option 66**—TFTP server name option. It specifies a TFTP server to be assigned to the client.
- **Option 67**—Bootfile name option. It specifies the bootfile name to be assigned to the client.
- **Option 121**—Classless route option. It specifies a list of classless static routes (the destination addresses in these static routes are classless) that the requesting client should add to its routing table.
- **Option 150**—TFTP server IP address option. It specifies the TFTP server IP address to be assigned to the client.

For more information about DHCP options, see RFC 2132.

Self-defined options

Some options, such as Option 43, have no unified definitions in RFC 2132.

Vendor-specific option (Option 43)

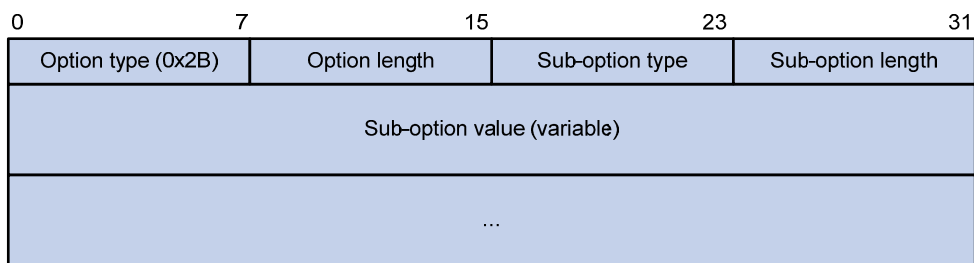
DHCP servers and clients use Option 43 to exchange vendor-specific configuration information. The client sends a request with Option 43, including a vendor string that identifies a vendor. Upon receiving the request, the DHCP server refers to the vendor-specific options table, and returns a response message with Option 43 to assign the appropriate vendor-specific information to the DHCP client.

The DHCP client can obtain the following information through Option 43:

- **ACS parameters**—Including the ACS URL, username, and password.
- **Service provider**—Identifier acquired by the CPE from the DHCP server and sent to the ACS for selecting vendor-specific configurations and parameters.
- **PXE server address**—For further obtaining the bootfile or other control information from the PXE server.

1. Format of Option 43

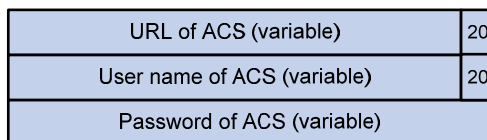
Figure 19 Format of Option 43



Network configuration parameters are carried in different sub-options of Option 43, as shown in Figure 19. The sub-option fields are described as follows:

- **Sub-option type**—Type of a sub-option. The field value can be 0x01, 0x02, or 0x80. 0x01 indicates an ACS parameter sub-option. 0x02 indicates a service provider identifier sub-option. 0x80 indicates a PXE server address sub-option.
 - **Sub-option length**—Length of a sub-option excluding the sub-option type and sub-option length fields.
 - **Sub-option value**—Value of a sub-option.
- ### 2. Format of the sub-option value field of Option 43
- As shown in Figure 20, the value field of the ACS parameter sub-option contains variable ACS URL, username, and password separated by spaces (0x20).

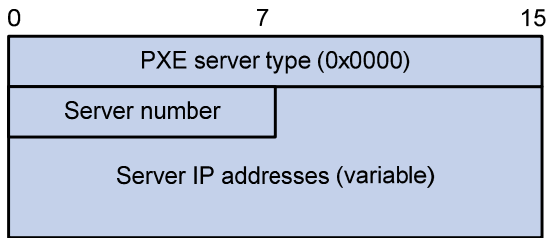
Figure 20 Format of the value field of the ACS parameter sub-option



- The value field of the service provider identifier sub-option contains the service provider identifier.

- **Figure 21** shows the format of the value field of the PXE server address sub-option. The value of the PXE server type can only be 0. The server number field indicates the number of PXE servers contained in the sub-option. The server IP addresses field contains the IP addresses of the PXE servers.

Figure 21 Format of the value field of the PXE server address sub-option



Relay agent option (Option 82)

Option 82 is the relay agent option in the option field of the DHCP message. It records the location information of the DHCP client. When a DHCP relay agent or DHCP snooping device receives a client’s request, it adds Option 82 to the request message and sends it to the server.

The administrator can locate the DHCP client to further implement security control and accounting. The Option 82 supporting server can also use such information to define individual assignment policies of IP address and other parameters for the clients.

Option 82 involves at most 255 sub-options. At least one sub-option must be defined. The DHCP relay agent supports two sub-options: sub-option 1 (Circuit ID) and sub-option 2 (Remote ID).

Option 82 has no unified definition. Its padding formats vary with vendors.

Use the following methods to configure Option 82:

- **User-defined method**—Manually specify the content of Option 82.
- **Non-user-defined method**—Pad Option 82 in the default normal or verbose format.

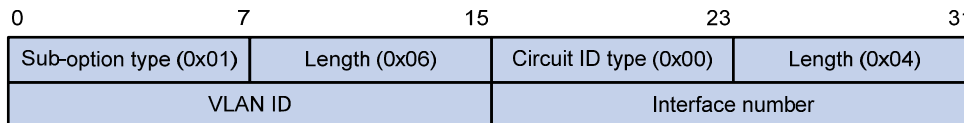
If you choose the second method, specify the code type for the sub-options as ASCII or HEX.

1. Normal padding format

The padding contents for sub-options in the normal padding format are as follows:

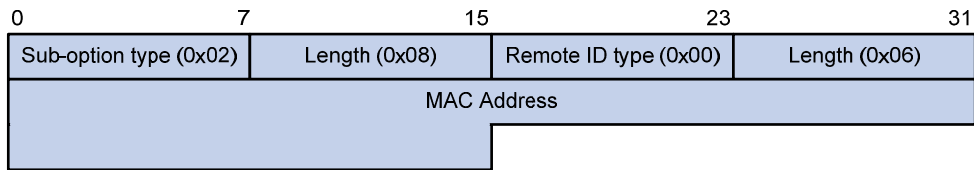
- **Sub-option 1**—Padded with the VLAN ID and interface number of the interface that received the client’s request. The value of the sub-option type is 1, and that of the circuit ID type is 0.

Figure 22 Sub-option 1 in normal padding format



- **Sub-option 2**—Padded with the MAC address of the DHCP relay agent interface or the MAC address of the DHCP snooping device that received the client’s request. The value of the sub-option type is 2, and that of the remote ID type is 0.

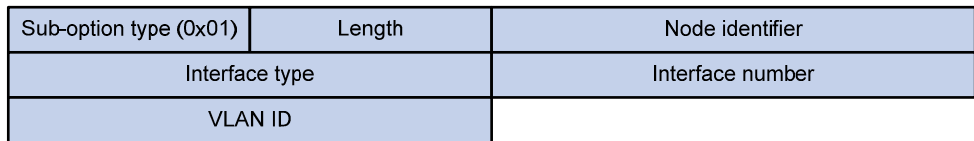
Figure 23 Sub-option 2 in normal padding format



2. Verbose padding format

- **Sub-option 1**—Padded with the user-specified access node identifier (ID of the switch that adds Option 82 in DHCP messages), and the type, number, and VLAN ID of the interface that received the client’s request. Its format is shown in [Figure 24](#).

Figure 24 Sub-option 1 in verbose padding format



The VLAN ID field has a fixed length of 2 bytes. All the other padding contents of sub-option 1 are length variable. See [Figure 24](#).

- **Sub-option 2**—Padded with the MAC address of the DHCP relay agent interface or the MAC address of the DHCP snooping device that received the client’s request. It has the same format as that in normal padding format. See [Figure 23](#).

Option 184

Option 184 is a reserved option, and parameters in the option can be defined as needed. The switch supports Option 184 carrying the voice related parameters, so a DHCP client with voice functions can get an IP address along with specified voice parameters from the DHCP server.

Option 184 involves the following sub-options:

- **Sub-option 1**—IP address of the primary network calling processor, which serves as the network calling control source and provides program downloads.
- **Sub-option 2**—IP address of the backup network calling processor that DHCP clients contacts when the primary one is unreachable.
- **Sub-option 3**—Voice VLAN ID and the result whether DHCP clients take this ID as the voice VLAN or not.
- **Sub-option 4**—Failover route that specifies the destination IP address and the called number that a SIP user uses to reach another SIP user when both the primary and backup calling processors are unreachable.

You must define the sub-option 1 to make other sub-options effective.

Protocols and standards

- RFC 2131, *Dynamic Host Configuration Protocol*
- RFC 2132, *DHCP Options and BOOTP Vendor Extensions*
- RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*
- RFC 3046, *DHCP Relay Agent Information Option*
- RFC 3442, *The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4*

DHCP server configuration

The DHCP server configuration is supported only on Layer 3 Ethernet interfaces, VLAN interfaces, Layer 3 aggregate interfaces, and loopback interfaces. The secondary IP address pool configuration is not supported on loopback interfaces.

The layer 3 Ethernet interface is an Ethernet interface operating in route mode. For more information about the operating mode of the Ethernet interface, see *Layer 2—LAN Switching Configuration Guide*.

Application environment

The DHCP server is well suited to networks where:

- Manual configuration and centralized management are difficult to implement.
- Many hosts need to acquire IP addresses dynamically. This can be because the number of hosts exceeds the number of assignable IP addresses, so it is impossible to assign a fixed IP address to each host. For example, an ISP has a limited number of host addresses.
- A few hosts need fixed IP addresses.

In addition to assigning IP addresses to DHCP clients on public networks, an MCE device serving as the DHCP server can also assign IP addresses to DHCP clients on private networks. The IP address ranges of public and private networks or those of private networks on the DHCP server cannot overlap each other. For more information about MCE, see *MPLS Configuration Guide*.

DHCP address pool

Address pool types

DHCP address pools include common and extended address pools.

- Common address pool: Supports both static binding and dynamic allocation.
- Extended address pool: Supports only dynamic allocation.

Common address pool structure

The common address pool database is organized as a tree. The root of the tree is the address pool for natural networks, branches are address pools for subnets, and leaves are addresses statically bound to clients. For the same level address pools, a previously configured pool has a higher selection priority than a new one.

At the very beginning, subnets inherit network parameters and clients inherit subnetwork parameters. Therefore, common parameters, for example a DNS server address, should be configured at the highest (network or subnet) level of the tree.

After establishment of the inheritance relationship, the new configuration at the higher level (parent) of the tree is:

- Inherited if the lower level (child) has no such configuration, or
- Overridden if the lower level (child) has such configuration.

The extended address pools on a DHCP server are independent of each other and no inheritance relationship exists among them.

IP address lease durations are not inherited.

Address pool selection principles

The DHCP server observes the following principles to select an address pool when assigning an IP address to a client:

1. If there is an address pool where an IP address is statically bound to the MAC address or ID of the client, the DHCP server selects this address pool and assigns the statically bound IP address to the client. For the configuration of this address pool, see [“Configuring static address allocation.”](#)
2. If the receiving interface has an extended address pool referenced, the DHCP server assigns an IP address from this address pool. If no IP address is available in the address pool, the DHCP server fails to assign an address to the client. For the configuration of such an address pool, see [“Configuring dynamic address allocation for an extended address pool.”](#)
3. Otherwise, the DHCP server selects the smallest common address pool that contains the IP address of the receiving interface (if the client and the server reside on the same subnet), or the smallest common address pool that contains the IP address specified in the giaddr field of the client’s request (if a DHCP relay agent is in-between). If no IP address is available in the address pool, the DHCP server fails to assign an address to the client because it cannot assign an IP address from the parent address pool to the client. For the configuration of such address pool, see [“Configuring dynamic address allocation.”](#)

For example, two common address pools, 1.1.1.0/24 and 1.1.1.0/25, are configured on the DHCP server. If the IP address of the interface receiving DHCP requests is 1.1.1.1/25, the DHCP server selects IP addresses for clients from address pool 1.1.1.0/25. If no IP address is available in the address pool, the DHCP server fails to assign addresses to clients. If the IP address of the interface receiving DHCP requests is 1.1.1.130/25, the DHCP server selects IP addresses for clients from the 1.1.1.0/24 address pool.

Keep the IP addresses for dynamic allocation within the subnet where the interface of the DHCP server or DHCP relay agent resides to avoid wrong IP address allocation.

IP address allocation sequence

A DHCP server assigns an IP address to a client according to the following sequence:

1. The IP address statically bound to the client’s MAC address or ID
2. The IP address that was ever assigned to the client
3. The IP address designated by the Option 50 field in a DHCP-DISCOVER message
4. The first assignable IP address found in an extended or a common address pool
5. The IP address that was a conflict or passed its lease duration

If no IP address is assignable, the server does not respond.

Option 50 is the requested IP address field in DHCP-DISCOVER messages. It is padded by the client to specify the IP address that the client wants to obtain. The contents to be padded depend on the client.

Configuration task list

| Task | Remarks |
|--|---|
| Configuring a DHCP server address pool | Required. |
| Enabling DHCP | Required. |
| Enabling the DHCP server on an interface | Required. |
| Applying an extended address pool on an interface | Required by the extended address pool configuration. When configuring a common address pool, ignore this task. |
| Configuring the DHCP server security functions | Optional. |
| Enabling Option 82 handling | Optional. |
| Specifying the threshold for sending trap messages | Optional. |

Configuring a DHCP server address pool

| Task | Remarks | | | |
|---|--|---------------------------------------|--|--|
| Creating a DHCP address pool | Required. | | | |
| Configuring an address allocation mode for a common address pool | <table border="0"> <tr> <td>Configuring static address allocation</td> <td rowspan="2">Required to configure either of the two for the common address pool configuration.</td> </tr> <tr> <td>Configuring dynamic address allocation</td> </tr> </table> | Configuring static address allocation | Required to configure either of the two for the common address pool configuration. | Configuring dynamic address allocation |
| Configuring static address allocation | Required to configure either of the two for the common address pool configuration. | | | |
| Configuring dynamic address allocation | | | | |
| Configuring dynamic address allocation for an extended address pool | Required for the extended address pool configuration. | | | |
| Configuring a domain name suffix for the client | Optional. | | | |
| Configuring DNS servers for the client | | | | |
| Configuring WINS servers and NetBIOS node type for the client | | | | |
| Configuring BIMS server information for the client | | | | |
| Configuring gateways for the client | | | | |
| Configuring Option 184 parameters for the client with voice service | | | | |
| Configuring the TFTP server and Bootfile name for the client | | | | |
| Specifying a server's IP address for the client | | | | |
| Configuring self-defined DHCP options | | | | |

Creating a DHCP address pool

When creating a DHCP address pool, specify it as a common address pool or an extended address pool.

To create a DHCP address pool:

| Step | Command | Remarks |
|---|--|--|
| 1. Enter system view. | system-view | — |
| 2. Create a DHCP address pool and enter its view. | dhcp server ip-pool <i>pool-name</i> [extended] | Required. No DHCP address pool is created by default. |

A common address pool and an extended address pool are different in address allocation mode configuration. Configurations of other parameters (such as the domain name suffix and DNS server address) for them are the same.

Configuring an address allocation mode for a common address pool

CAUTION:

Configure either the static binding or dynamic address allocation for a common address pool as needed.

You must specify an address range for the dynamic address allocation. A static binding is a special address pool containing only one IP address.

Configuring static address allocation

Some DHCP clients such as a WWW server need fixed IP addresses. create a static binding of a client's MAC or ID to IP address in the DHCP address pool.

When the client with the MAC address or ID requests an IP address, the DHCP server finds the IP address from the binding for the client.

A DHCP address pool now supports only one static binding, which can be a MAC-to-IP or ID-to-IP binding.

To configure a static binding in a common address pool:

| Step | Command | Remarks |
|---|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter common address pool view. | dhcp server ip-pool <i>pool-name</i> | — |
| 3. Specify the IP address of the binding. | static-bind ip-address <i>ip-address</i> [<i>mask-length</i> mask <i>mask</i>] | Required. No IP addresses are statically bound by default. |
| 4. Specify the MAC address or client ID. | Specify the MAC address static-bind mac-address <i>mac-address</i> | Required to configure either of the two. Neither is bound statically by default. |
| | Specify the client ID static-bind client-identifier <i>client-identifier</i> | |

| Step | Command | Remarks |
|---|---|--|
| 5. Specify the lease duration for the IP address. | expired { day <i>day</i> [hour <i>hour</i> [minute <i>minute</i> [second <i>second</i>]]] unlimited } | Optional. By default, the lease duration of an IP address is unlimited. |

Use **static-bind ip-address** together with **static-bind mac-address** or **static-bind client-identifier** to accomplish a static binding configuration.

In a DHCP address pool, if you execute **static-bind mac-address** before **static-bind client-identifier**, the latter overwrites the former and vice versa.

If you use **static-bind ip-address**, **static-bind mac-address**, or **static-bind client-identifier** repeatedly in the DHCP address pool, the new configuration overwrites the previous one.

The IP address of the static binding cannot be an interface address of the DHCP server. Otherwise, an IP address conflict can occur and the bound client cannot obtain an IP address correctly.

The ID of the static binding must be identical to the ID displayed by using **display dhcp client verbose** on the client. Otherwise, the client cannot obtain an IP address.

When the switch serves as a DHCP client or BOOTP client, you need to configure the static binding of the DHCP client's ID to IP address, or the static binding of the BOOTP client's MAC to IP address on the DHCP server; otherwise, the DHCP or BOOTP client cannot obtain a static IP address.

If the interfaces on a DHCP client share the same MAC address, you need to specify the client ID, rather than MAC address, in a static binding to identify the requesting interface; otherwise, the client can fail to obtain an IP address.

Configuring dynamic address allocation

For dynamic address allocation, you must configure a DHCP address pool, specify one and only one address range for the pool, and specify the lease duration. A DHCP address pool can have only one lease duration.

To avoid address conflicts, configure the DHCP server to exclude IP addresses used by the gateway or FTP server from dynamic allocation.

To configure dynamic address allocation for a common address pool:

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter common address pool view. | dhcp server ip-pool <i>pool-name</i> | — |
| 3. Specify a subnet. | network <i>network-address</i> [<i>mask-length</i> mask <i>mask</i>] | Required. Not specified by default. |
| 4. Specify the address lease duration. | expired { day <i>day</i> [hour <i>hour</i> [minute <i>minute</i> [second <i>second</i>]]] unlimited } | Optional. One day by default. |
| 5. Return to system view. | quit | — |
| 6. Exclude IP addresses from automatic allocation. | dhcp server forbidden-ip <i>low-ip-address</i> [<i>high-ip-address</i>] | Optional. Except IP addresses of the DHCP server interfaces, all addresses in the DHCP address pool are assignable by default. |

In common address pool view, using **network** repeatedly overwrites the previous configuration.

After you exclude IP addresses from automatic allocation by using **dhcp server forbidden-ip**, neither a common address pool nor an extended address pool can assign these IP addresses through dynamic address allocation.

Using **dhcp server forbidden-ip** repeatedly can exclude multiple IP address ranges from allocation.

Configuring dynamic address allocation for an extended address pool

Extended address pools support only dynamic address allocation.

When configuring an extended address pool, you need to specify:

- Assignable IP address range
- Mask

After the assignable IP address range and the mask are specified, the address pool becomes valid.

To configure dynamic address allocation for an extended address pool:

| Step | Command | Remarks |
|---|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter extended address pool view. | dhcp server ip-pool pool-name extended | — |
| 3. Specify the IP address range. | network ip range min-address max-address | Required. Not specified by default. |
| 4. Specify the IP address mask. | network mask mask | Required. Not specified by default. |
| 5. Specify the IP address range for the DHCP clients of a specified vendor. | vendor-class-identifier hex-string<1-255> ip range min-address max-address | Optional. Not configured by default. |
| 6. Specify the address lease duration. | expired { day day [hour hour [minute minute [second second]]] unlimited } | Optional. One day by default. |
| 7. Exclude IP addresses from dynamic allocation. | forbidden-ip ip-address<1-8> | Optional. Except IP addresses of the DHCP server interfaces, all addresses in the DHCP address pool are assignable by default. |

Excluded IP addresses specified with **forbidden-ip** in DHCP address pool view are not assignable in the current extended address pool, but are assignable in other address pools.

Configuring a domain name suffix for the client

Specify a domain name suffix in each DHCP address pool on the DHCP server to provide the clients with the domain name suffix. With this suffix assigned, the client only needs to enter part of a domain name,

and the system adds the domain name suffix for name resolution. For more information about DNS, see the chapter “IPv4 DNS configuration.”

To configure a domain name suffix in the DHCP address pool:

| Step | Command | Remarks |
|----------------------------------|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Specify a domain name suffix. | domain-name <i>domain-name</i> | Required. Not specified by default. |

Configuring DNS servers for the client

A DHCP client contacts a DNS server to resolve names. specify up to eight DNS servers in the DHCP address pool.

To configure DNS servers in the DHCP address pool:

| Step | Command | Remarks |
|----------------------------------|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Specify DNS servers. | dns-list <i>ip-address&<1-8></i> | Required. Not specified by default. |

Configuring WINS servers and NetBIOS node type for the client

A Microsoft DHCP client using NetBIOS protocol contacts a WINS server for name resolution. Therefore, the DHCP server should assign a WINS server address when assigning an IP address to the client.

Specify up to eight WINS servers in a DHCP address pool.

You need to specify in a DHCP address pool a NetBIOS node type for the client to approach name resolution. There are four NetBIOS node types:

- **B (broadcast)-node**—The b-node client sends the destination name in a broadcast message. The destination returns its IP address to the client after receiving the message.
- **P (peer-to-peer)-node**—The p-node client sends the destination name in a unicast message to the WINS server, and the WINS server returns the destination IP address.
- **M (mixed)-node**—A combination of broadcast first and peer-to-peer second. The m-node client broadcasts the destination name, if no response is received, then unicasts the destination name to the WINS server to get the destination IP address.
- **H (hybrid)-node**—A combination of peer-to-peer first and broadcast second. The h-node client unicasts the destination name to the WINS server, and if no response is received, broadcasts it to get the destination IP address.

To configure WINS servers and NetBIOS node type in the DHCP address pool:

| Step | Command | Remarks |
|--------------------------------------|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Specify WINS server IP addresses. | nbns-list <i>ip-address</i> &<1-8> | Required (optional for b-node). No address is specified by default. |
| 4. Specify the NetBIOS node type. | netbios-type { b-node h-node m-node p-node } | Required. Not specified by default. |

If b-node is specified for the client, you do not need to specify any WINS server address.

Configuring BIMS server information for the client

Some DHCP clients perform regular software update and backup by using configuration files obtained from a BIMS server. Therefore, the DHCP server must offer these DHCP clients the BIMS server IP address, port number, shared key from the DHCP address pool.

To configure the BIMS server IP address, port number, and shared key in the DHCP address pool:

| Step | Command | Remarks |
|---|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Specify the BIMS server IP address, port number, and shared key. | bims-server ip <i>ip-address</i> [port <i>port-number</i>] sharekey <i>key</i> | Required. Not specified by default. |

Configuring gateways for the client

Specify up to eight gateways in a DHCP address pool.

To configure the gateways in the DHCP address pool:

| Step | Command | Remarks |
|----------------------------------|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Specify gateways. | gateway-list <i>ip-address</i> &<1-8> | Required. No gateway is specified by default. |

Configuring Option 184 parameters for the client with voice service

To assign voice calling parameters along with an IP address to DHCP clients with voice service, you need to configure Option 184 on the DHCP server. For more information about Option 184, see “[Option 184.](#)”

If Option 55 in the request from a DHCP client contains Option 184, the DHCP server returns parameters specified in Option 184 to the client. The client then can initiate a call using parameters in Option 184.

To configure option 184 parameters in the DHCP address pool:

| Step | Command | Remarks |
|---|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Specify the IP address of the primary network calling processor. | voice-config ncp-ip <i>ip-address</i> | Required. Not specified by default. |
| 4. Specify the IP address of the backup network calling processor. | voice-config as-ip <i>ip-address</i> | Optional. Not specified by default. |
| 5. Configure the voice VLAN. | voice-config voice-vlan <i>vlan-id</i> { disable enable } | Optional. Not configured by default. |
| 6. Specify the failover IP address and dialer string. | voice-config fail-over <i>ip-address dialer-string</i> | Optional. No failover IP address or dialer string is specified by default. |

Specify an IP address for the network calling processor before performing other configurations.

Configuring the TFTP server and Bootfile name for the client

For the DHCP server to support client auto-configuration, you must specify the IP address or name of a TFTP server and the bootfile name in the DHCP address pool. You do not need to perform any configuration on the DHCP client.

1. When a router starts up without loading any configuration file, the system sets an active interface (such as the interface of the default VLAN or a Layer 3 Ethernet interface) as the DHCP client to request from the DHCP server for parameters, such as an IP address and name of a TFTP server, and the bootfile name.
2. After getting related parameters, the DHCP client sends a TFTP request to obtain the configuration file from the specified TFTP server for system initialization. If the client cannot get such parameters, it performs system initialization without loading any configuration file.

When Option 55 in the client's request contains parameters of Option 66, Option 67, or Option 150, the DHCP server returns the IP address or name of the specified TFTP server, and bootfile name to the client.

To configure the IP address and name of the TFTP server and the bootfile name in the DHCP address pool:

| Step | Command | Remarks |
|---|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Specify the TFTP server. | tftp-server ip-address <i>ip-address</i> | Required to use either command. |
| 4. Specify the name of the TFTP server. | tftp-server domain-name <i>domain-name</i> | Not specified by default. |
| 5. Specify the bootfile name. | bootfile-name <i>bootfile-name</i> | Required. Not specified by default. |

Specifying a server's IP address for the client

Some DHCP clients need to obtain boot configuration information from a server after they get configuration information, such as IP addresses, from the DHCP server. specify the IP address of that server in each address pool of the DHCP server. The DHCP server sends the server's IP address to DHCP clients along with other configuration information.

To specify the IP address of a server:

| Step | Command | Remarks |
|--|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Specify the IP address of a server. | next-server <i>ip-address</i> | Required. Not specified by default. |

Configuring self-defined DHCP options

CAUTION:

Be cautious when configuring self-defined DHCP options because such configuration can affect the operation of DHCP.

By configuring self-defined DHCP options, you can

- Define new DHCP options. New configuration options come out with DHCP development. To support these new options, add them into the attribute list of the DHCP server.
- Define existing DHCP options. Vendors use Option 43 to define options that have no unified definitions in RFC 2132. The self-defined DHCP option enables DHCP clients to obtain vendor-specific information.
- Extend existing DHCP options. When the current DHCP options cannot meet the customers' requirements (for example, you cannot use **dns-list** to configure more than eight DNS server addresses), configure a self-defined option for extension.

To configure a self-defined DHCP option in the DHCP address pool:

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter DHCP address pool view. | dhcp server ip-pool <i>pool-name</i> [extended] | — |
| 3. Configure a self-defined DHCP option. | option code { ascii <i>ascii-string</i> hex <i>hex-string</i> &<1-16> ip-address <i>ip-address</i> &<1-8> } | Required. No DHCP option is configured by default. |

Table 2 Description of common options

| Option | Option name | Corresponding command | Command parameter |
|--------|--|-----------------------|-------------------|
| 3 | Router Option | gateway-list | ip-address |
| 6 | Domain Name Server Option | dns-list | ip-address |
| 15 | Domain Name | domain-name | ascii |
| 44 | NetBIOS over TCP/IP Name Server Option | nbns-list | ip-address |
| 46 | NetBIOS over TCP/IP Node Type Option | netbios-type | hex |
| 66 | TFTP server name | tftp-server | ascii |
| 67 | Bootfile name | bootfile-name | ascii |
| 43 | Vendor Specific Information | — | hex |

Enabling DHCP

Enable DHCP before performing other configurations.

To enable DHCP:

| Step | Command | Remarks |
|-----------------------|--------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable DHCP. | dhcp enable | Required. Disabled by default. |

Enabling the DHCP server on an interface

With the DHCP server enabled on an interface, upon receiving a client's request, the DHCP server assigns an IP address from its address pool to the DHCP client.

To enable the DHCP server on an interface:

| Step | Command | Remarks |
|--|--|----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Enable the DHCP server on an interface. | dhcp select server global-pool [subaddress] | Optional. Enabled by default. |

If a DHCP relay agent exists between the DHCP server and client, the DHCP server, regardless of whether the **subaddress** keyword is used, selects an IP address from the address pool containing the primary IP address of the DHCP relay agent's interface (connected to the client) for a requesting client.

When the DHCP server and client are on the same subnet:

- With the keyword **subaddress** specified, the DHCP server preferably assigns an IP address from an address pool that resides on the same subnet as the primary IP address of the server interface (connecting to the client). If the address pool contains no assignable IP address, the server assigns an IP address from an address pool that resides on the same subnet as the secondary IP addresses of the server interface. If the interface has multiple secondary IP addresses, each address pool is tried in turn for address allocation.
- Without the keyword **subaddress** specified, the DHCP server can only assign an IP address from the address pool that resides on the same subnet as the primary IP address of the server interface.

Applying an extended address pool on an interface

After you create an extended address pool and apply it on an interface, the DHCP server, upon receiving a client's request on the interface, attempts to assign the client the statically bound IP address first and then an IP address from the specified address pool. If no IP address is available, address allocation fails, and the DHCP server does not assign the client any IP address from other address pools.

To apply an extended address pool on an interface:

| Step | Command | Remarks |
|---|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Apply an extended address pool on the interface. | dhcp server apply ip-pool pool-name | Optional. By default, the DHCP server has no extended address pool applied on its interface, and assigns an IP address from a common address pool to a requesting client. |

Only an extended address pool can be applied on the interface. The address pool to be referenced must already exist.

Configuring the DHCP server security functions

Configuration prerequisites

Before performing this configuration, complete the following configurations on the DHCP server:

- Enable DHCP.
- Configure the DHCP address pool.

Enabling unauthorized DHCP server detection

Unauthorized DHCP servers on a network can assign wrong IP addresses to DHCP clients.

With unauthorized DHCP server detection enabled, the DHCP server checks whether a DHCP request contains Option 54 (Server Identifier Option). If yes, the DHCP server records the IP address in the option, which is the IP address of the DHCP server that assigned an IP address to the DHCP client and records the receiving interface. The administrator can use this information to check for unauthorized DHCP servers.

To enable unauthorized DHCP server detection:

| Step | Command | Remarks |
|---|---------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable unauthorized DHCP server detection. | dhcp server detect | Required. Disabled by default. |

With the unauthorized DHCP server detection enabled, the switch puts logs each detected a record once. The administrator can use the log information to find unauthorized DHCP servers.

Configuring IP address conflict detection

With IP address conflict detection enabled, the DHCP server pings each IP address to be assigned using ICMP. If the server receives a response within the specified period, the server selects and ping another IP address; otherwise, the server pings the IP addresses once again until the specified number of ping packets are sent. If still no response is received, the server assigns the IP address to the requesting client (the DHCP client probes the IP address by sending gratuitous ARP packets).

To configure IP address conflict detection:

| Step | Command | Remarks |
|--|--|--|
| 1. Enter system view. | system-view | — |
| 2. Specify the number of ping packets. | dhcp server ping packets <i>number</i> | Optional. One ping packet by default. The value 0 indicates that no ping operation is performed. |
| 3. Configure a timeout waiting for ping responses. | dhcp server ping timeout <i>milliseconds</i> | Optional. 500 ms by default. The value 0 indicates that no ping operation is performed. |

Enabling Option 82 handling

With Option 82 handling enabled, when the DHCP server receives a request with Option 82, it adds Option 82 into the response.

If the server is configured to ignore Option 82, it assigns an IP address to the client without adding Option 82 in the response message.

Configuration prerequisites

Before performing this configuration, complete the following configuration on the DHCP server:

- Enable DHCP
- Configure the DHCP address pool

Enable Option 82 handling

| Step | Command | Remarks |
|---|---|----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable the server to handle Option 82. | dhcp server relay information enable | Optional. Enabled by default. |

Supporting Option 82 requires configuring both the DHCP server and relay agent (or the switch enabled with DHCP snooping). For more information, see “[DHCP relay agent configuration](#)” and “[DHCP snooping configuration](#)”

Specifying the threshold for sending trap messages

Configuration prerequisites

Before performing the configuration, use **snmp-agent target-host** to specify the destination address of the trap messages. For more information about the command, see the *Network Management and Monitoring Command Reference*.

Configuration procedure

A DHCP server sends trap messages to the network management server when one of the following items reaches the specified threshold:

- The ratio of successfully allocated IP addresses to received DHCP requests.
- The average IP address utilization of the address pool.
- The maximum IP address utilization of the address pool.

Trap messages help network administrators know the latest usage information of the DHCP server.

To specify the threshold for sending trap messages:

| Step | Command | Remarks |
|--|---|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Specify the threshold for sending trap messages to the network management server. | dhcp server threshold { allocated-ip <i>threshold-value</i> average-ip-use <i>threshold-value</i> max-ip-use <i>threshold-value</i> } | Optional. Disabled by default. |

Displaying and maintaining the DHCP server

| Task | Command | Remarks |
|---|---|-----------------------|
| Display information about IP address conflicts. | display dhcp server conflict { all ip <i>ip-address</i> } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display information about lease expiration. | display dhcp server expired { all ip <i>ip-address</i> pool [<i>pool-name</i>] } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display information about assignable IP addresses. | display dhcp server free-ip [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display IP addresses excluded from automatic allocation in the DHCP address pool. | display dhcp server forbidden-ip [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display information about bindings. | display dhcp server ip-in-use { all ip <i>ip-address</i> pool [<i>pool-name</i>] } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display information about DHCP server statistics. | display dhcp server statistics [{ begin exclude include } <i>regular-expression</i>] | Available in any view |

| Task | Command | Remarks |
|---|---|------------------------|
| Display tree organization information of address pools. | <code>display dhcp server tree { all pool [pool-name] } [{ begin exclude include } regular-expression]</code> | Available in any view |
| Clear information about IP address conflicts. | <code>reset dhcp server conflict { all ip ip-address }</code> | Available in user view |
| Clear information about dynamic bindings. | <code>reset dhcp server ip-in-use { all ip ip-address pool [pool-name] }</code> | Available in user view |
| Clear information about DHCP server statistics. | <code>reset dhcp server statistics</code> | Available in user view |

Using **save** does not save DHCP server lease information. Therefore, when the system boots up or **reset dhcp server ip-in-use** is executed, no lease information is available in the configuration file. The server denies the request for lease extension from a client and the client must request an IP address again.

DHCP server configuration examples

DHCP networking involves two types:

- The DHCP server and client are on the same subnet and exchange messages directly.
- The DHCP server and client are not on the same subnet and they communicate with each other via a DHCP relay agent.

The DHCP server configuration for the two types is the same.

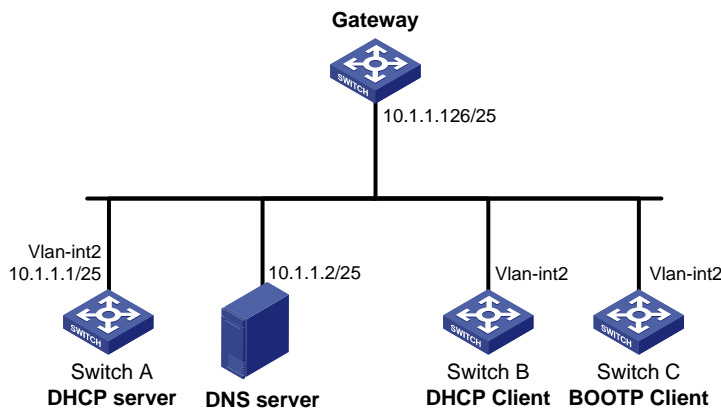
Static IP address assignment configuration example

Network requirements

As shown in [Figure 25](#), Switch B (DHCP client) and Switch C (BOOTP client) obtain the static IP address, DNS server address, and gateway address from Switch A (DHCP server).

The client ID of VLAN-interface 2 on Switch B is 3030-3066-2e65-3234-392e-3830-3530-2d56-6c61-6e2d-696e-7465-7266-6163-6532. The MAC address of VLAN-interface 2 on Switch C is 000f-e249-8050.

Figure 25 Network diagram for static IP address assignment



Configuration procedure

1. Configure the IP address of VLAN-interface 2 on Switch A.

```
<SwitchA> system-view
[SwitchA] interface vlan-interface 2
[SwitchA-Vlan-interface2] ip address 10.1.1.1 25
[SwitchA-Vlan-interface2] quit
```

2. Configure the DHCP server

Enable DHCP.

```
[SwitchA] dhcp enable
```

Enable the DHCP server on VLAN-interface 2.

```
[SwitchA] interface vlan-interface 2
[SwitchA-Vlan-interface2] dhcp select server global-pool
[SwitchA-Vlan-interface2] quit
```

Create DHCP address pool 0, configure a static binding, DNS server and gateway in it.

```
[SwitchA] dhcp server ip-pool 0
[SwitchA-dhcp-pool-0] static-bind ip-address 10.1.1.5
[SwitchA-dhcp-pool-0] static-bind client-identifier 3030-3066-2e65-3234-392e-3830-3530-
2d56-6c61-6e2d-696e-7465-7266-6163-6532
[SwitchA-dhcp-pool-0] dns-list 10.1.1.2
[SwitchA-dhcp-pool-0] gateway-list 10.1.1.126
[SwitchA-dhcp-pool-0] quit
```

Create DHCP address pool 1, configure a static binding, DNS server and gateway in it.

```
[SwitchA] dhcp server ip-pool 1
[SwitchA-dhcp-pool-1] static-bind ip-address 10.1.1.6
[SwitchA-dhcp-pool-1] static-bind mac-address 000f-e249-8050
[SwitchA-dhcp-pool-1] dns-list 10.1.1.2
[SwitchA-dhcp-pool-1] gateway-list 10.1.1.126
```

3. Verification

After the preceding configuration is complete, Switch B can obtain IP address 10.1.1.5 and other network parameters, and Switch C can obtain IP address 10.1.1.6 and other network parameters from Switch A. Use **display dhcp server ip-in-use** on the DHCP server to view the IP addresses assigned to the clients.

Dynamic IP address assignment configuration example

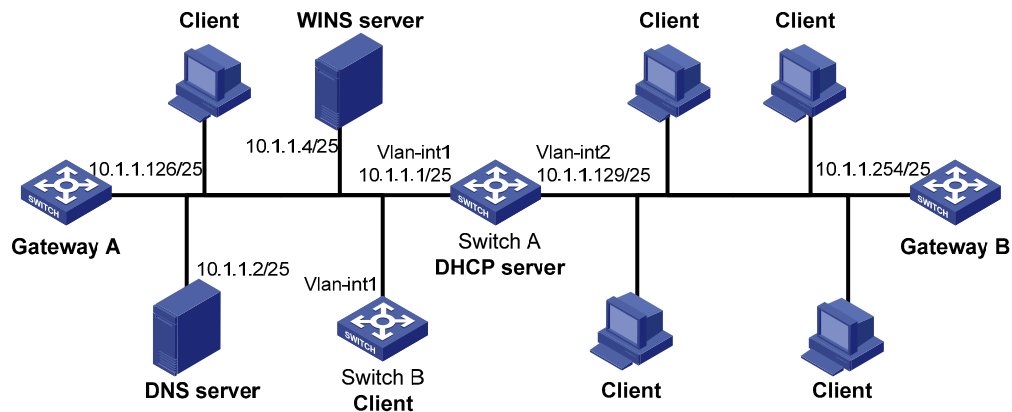
Network requirements

- As shown in [Figure 26](#), the DHCP server (Switch A) assigns IP addresses to clients in subnet 10.1.1.0/24, which is subnetted into 10.1.1.0/25 and 10.1.1.128/25.
- The IP addresses of VLAN-interfaces 1 and 2 on Switch A are 10.1.1.1/25 and 10.1.1.129/25 respectively.
- In address pool 10.1.1.0/25, configure the address lease duration as ten days and twelve hours, domain name suffix aabbcc.com, DNS server address 10.1.1.2/25, gateway 10.1.1.126/25, and WINS server 10.1.1.4/25.
- In address pool 10.1.1.128/25, configure the address lease duration as five days, domain name suffix aabbcc.com, DNS server address 10.1.1.2/25, and gateway address 10.1.1.254/25, and there is no WINS server address.

- The domain name and DNS server address on subnets 10.1.1.0/25 and 10.1.1.128/25 are the same. Therefore, the domain name suffix and DNS server address can only be configured for subnet 10.1.1.0/24. Subnet 10.1.1.128/25 can inherit the configuration of subnet 10.1.1.0/24.

In this example, the number of requesting clients connected to VLAN-interface 1 should be less than 122, and that of clients connected to VLAN-interface 2 should be less than 124.

Figure 26 DHCP network diagram



Configuration procedure

1. Specify IP addresses for VLAN interfaces (omitted).
2. Configure the DHCP server

Enable DHCP.

```
<SwitchA> system-view
[SwitchA] dhcp enable
```

Enable the DHCP server on VLAN-interface 1 and VLAN-interface 2.

```
[SwitchA] interface vlan-interface 1
[SwitchA-Vlan-interface1] dhcp select server global-pool
[SwitchA-Vlan-interface1] quit
[SwitchA] interface vlan-interface 2
[SwitchA-Vlan-interface2] dhcp select server global-pool
[SwitchA-Vlan-interface2] quit
```

Exclude IP addresses (addresses of the DNS server, WINS server and gateways).

```
[SwitchA] dhcp server forbidden-ip 10.1.1.2
[SwitchA] dhcp server forbidden-ip 10.1.1.4
[SwitchA] dhcp server forbidden-ip 10.1.1.126
[SwitchA] dhcp server forbidden-ip 10.1.1.254
```

```

# Configure DHCP address pool 0 (subnet, client domain name suffix, and DNS server address).
[SwitchA] dhcp server ip-pool 0
[SwitchA-dhcp-pool-0] network 10.1.1.0 mask 255.255.255.0
[SwitchA-dhcp-pool-0] domain-name aabbcc.com
[SwitchA-dhcp-pool-0] dns-list 10.1.1.2
[SwitchA-dhcp-pool-0] quit

# Configure DHCP address pool 1 (subnet, gateway, lease duration, and WINS server).
[SwitchA] dhcp server ip-pool 1
[SwitchA-dhcp-pool-1] network 10.1.1.0 mask 255.255.255.128
[SwitchA-dhcp-pool-1] gateway-list 10.1.1.126
[SwitchA-dhcp-pool-1] expired day 10 hour 12
[SwitchA-dhcp-pool-1] nbns-list 10.1.1.4
[SwitchA-dhcp-pool-1] quit

# Configure DHCP address pool 2 (subnet, gateway, and lease duration).
[SwitchA] dhcp server ip-pool 2
[SwitchA-dhcp-pool-2] network 10.1.1.128 mask 255.255.255.128
[SwitchA-dhcp-pool-2] expired day 5
[SwitchA-dhcp-pool-2] gateway-list 10.1.1.254

```

3. Verification

After the preceding configuration is complete, clients on networks 10.1.1.0/25 and 10.1.1.128/25 can obtain IP addresses on the corresponding network and other network parameters from Switch A. Use **display dhcp server ip-in-use** on the DHCP server to view the IP addresses assigned to the clients.

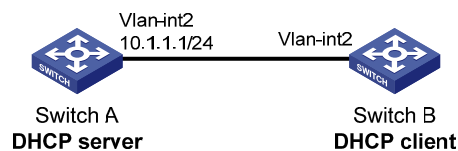
Self-defined option configuration example

Network requirements

As shown in [Figure 27](#), the DHCP client (Switch B) obtains an IP address and PXE server addresses from the DHCP server (Switch A). The IP address belongs to subnet 10.1.1.0/24. The PXE server addresses are 1.2.3.4 and 2.2.2.2.

The DHCP server assigns PXE server addresses to DHCP clients through Option 43, a self-defined option. The format of Option 43 and that of the PXE server address sub-option are shown in [Figure 19](#) and [Figure 21](#), respectively. The value of Option 43 configured on the DHCP server in this example is 80 0B 00 00 02 01 02 03 04 02 02 02 02. The number 80 is the value of the sub-option type. The number 0B is the value of the sub-option length. The numbers 00 00 are the value of the PXE server type. The number 02 indicates the number of servers. The numbers 01 02 03 04 02 02 02 02 indicate that the PXE server addresses are 1.2.3.4 and 2.2.2.2.

Figure 27 Network diagram for self-defined option configuration (a switch as the DHCP server)



Configuration procedure

1. Specify IP addresses for the interfaces (omitted).

2. Configure the DHCP server

Enable DHCP.

```
<SwitchA> system-view  
[SwitchA] dhcp enable
```

Enable the DHCP server on VLAN-interface 2.

```
[SwitchA] interface vlan-interface 2  
[SwitchA-Vlan-interface2] dhcp select server global-pool  
[SwitchA-Vlan-interface2] quit
```

Configure DHCP address pool 0.

```
[SwitchA] dhcp server ip-pool 0  
[SwitchA-dhcp-pool-0] network 10.1.1.0 mask 255.255.255.0  
[SwitchA-dhcp-pool-0] option 43 hex 80 0B 00 00 02 01 02 03 04 02 02 02 02
```

3. Verification

After the preceding configuration is complete, Switch B can obtain its IP address on 10.1.1.0/24 and PXE server addresses from the Switch A. use the **display dhcp server ip-in-use** command on the DHCP server to view the IP addresses assigned to the clients.

Troubleshooting DHCP server configuration

Symptom

A client's IP address obtained from the DHCP server conflicts with another IP address.

Analysis

A host on the subnet can have the same IP address.

Solution

1. Disable the client's network adapter or disconnect the client's network cable. Ping the client's IP address from another host to check whether there is a host using the same IP address.
2. If a ping response is received, the IP address has been manually configured on the host. Execute **dhcp server forbidden-ip** on the DHCP server to exclude the IP address from dynamic allocation.
3. Enable the network adapter or connect the network cable. Release the IP address and obtain another one on the client. Take WINDOW XP as an example, run **cmd** to enter DOS window. Enter **ipconfig/release** to relinquish the IP address and then **ipconfig/renew** to obtain another IP address.

DHCP relay agent configuration

The DHCP relay agent configuration is only supported on Layer 3 Ethernet interfaces, and VLAN interfaces.

Application environment

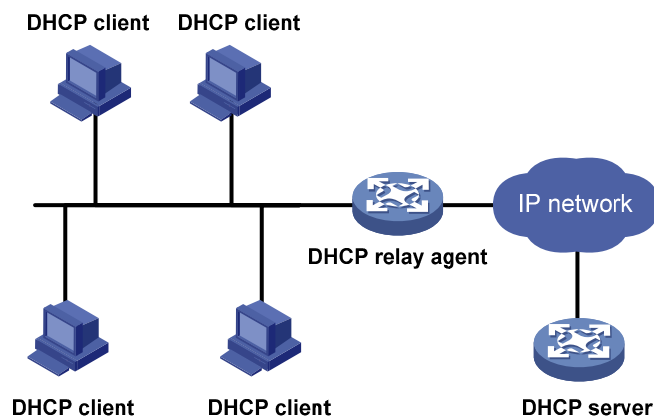
Via a relay agent, DHCP clients communicate with a DHCP server on another subnet to obtain configuration parameters. Thus, DHCP clients on different subnets can contact the same DHCP server rather than having a DHCP server on each subnet.

An MCE device serving as the DHCP relay agent can forward DHCP packets not only between a DHCP server and clients on a private network, but also between a DHCP server and clients on a public network. The IP address ranges of the public and private networks or those of private networks cannot overlap each other. For more information about MCE, see *MPLS Configuration Guide*.

Fundamentals

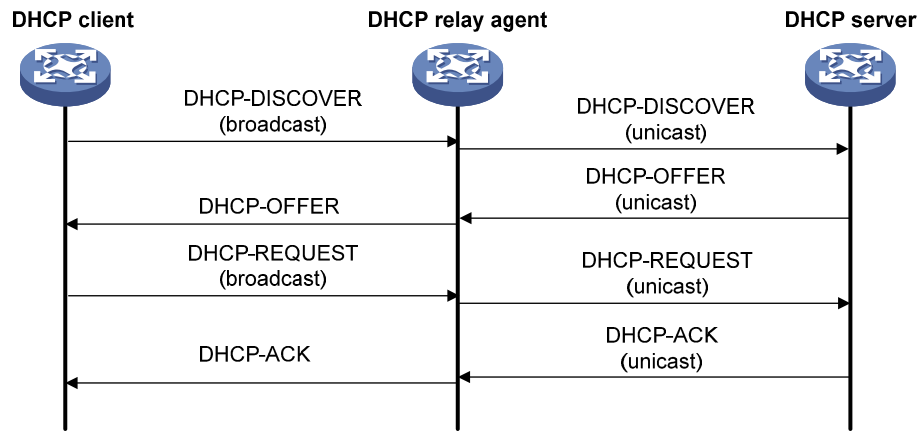
Figure 28 shows a typical application of the DHCP relay agent.

Figure 28 DHCP relay agent application



No matter whether a relay agent exists or not, the DHCP server and client interact with each other in a similar way (see "[DHCP overview](#)").

Figure 29 DHCP relay agent work process



1. After receiving a DHCP-DISCOVER or DHCP-REQUEST broadcast message from a DHCP client, the DHCP relay agent fills the giaddr field of the message with its IP address and forwards the message to the designated DHCP server in unicast mode.
2. Based on the giaddr field, the DHCP server returns an IP address and other configuration parameters to the relay agent, and the relay agent conveys them to the client.

DHCP relay agent support for Option 82

Option 82 records the location information of the DHCP client, letting the administrator locate the DHCP client for security control and accounting purposes. For more information, see “[Relay agent option \(Option 82\)](#).”

If the DHCP relay agent supports Option 82, it handles a client’s request according to the contents defined in Option 82, if any. The handling strategies are described in [Table 3](#).

If a reply returned by the DHCP server contains Option 82, the DHCP relay agent removes the Option 82 before forwarding the reply to the client.

Table 3 Handling strategies of the DHCP relay agent

| If a client’s requesting message has... | Handling strategy | Padding format | The DHCP relay agent will... |
|---|-------------------|----------------|---|
| | Drop | Random | Drop the message. |
| | Keep | Random | Forward the message without changing Option 82. |
| Option 82 | | normal | Forward the message after replacing the original Option 82 with the Option 82 padded in normal format. |
| | Replace | verbose | Forward the message after replacing the original Option 82 with the Option 82 padded in verbose format. |
| | | user-defined | Forward the message after replacing the original Option 82 with the user-defined Option 82. |

| If a client's requesting message has... | Handling strategy | Padding format | The DHCP relay agent will... |
|---|-------------------|----------------|--|
| no Option 82 | — | normal | Forward the message after adding the Option 82 padded in normal format. |
| | — | verbose | Forward the message after adding the Option 82 padded in verbose format. |
| | — | user-defined | Forward the message after adding the user-defined Option 82. |

Configuration task list

| Task | Remarks |
|--|-----------|
| Enabling DHCP | Required. |
| Enabling the DHCP relay agent on an interface | Required. |
| Correlating a DHCP server group with a relay agent interface | Required. |
| Configuring the DHCP relay agent security functions | Optional. |
| Enabling offline detection | Optional. |
| Configuring the DHCP relay agent to release an IP address | Optional. |
| Configuring the DHCP relay agent to support Option 82 | Optional. |

Enabling DHCP

Enable DHCP before performing other configurations related to the DHCP relay agent.

To enable DHCP:

| Step | Command | Remarks |
|-----------------------|--------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable DHCP. | dhcp enable | Required. Disabled by default. |

Enabling the DHCP relay agent on an interface

With the DHCP relay agent enabled, an interface forwards incoming DHCP requests to a DHCP server for address allocation.

To enable the DHCP relay agent on an interface:

| Step | Command | Remarks |
|--|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Enable the DHCP relay agent on the current interface. | dhcp select relay | Required. With DHCP enabled, interfaces work in the DHCP server mode. |

The IP address pool containing the IP address of the DHCP relay agent enabled interface must be configured on the DHCP server. Otherwise, DHCP clients cannot obtain correct IP addresses.

Correlating a DHCP server group with a relay agent interface

To improve reliability, specify several DHCP servers as a group on the DHCP relay agent and correlate a relay agent interface with the server group. When the interface receives request messages from clients, the relay agent forwards them to all DHCP servers of the group.

To correlate a DHCP server group with a relay agent interface:

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Create a DHCP server group and add a server into the group. | dhcp relay server-group group-id ip ip-address | Required. Not created by default. |
| 3. Enter interface view. | interface interface-type interface-number | — |
| 4. Correlate the DHCP server group with the current interface. | dhcp relay server-select group-id | Required. By default, no interface is correlated with any DHCP server group. |

By executing **dhcp relay server-group** repeatedly, specify up to eight DHCP server addresses for each DHCP server group.

The IP addresses of DHCP servers and those of relay agent's interfaces that connect DHCP clients cannot be on the same subnet. Otherwise, the client cannot obtain an IP address.

A DHCP server group can correlate with one or multiple DHCP relay agent interfaces, while a relay agent interface can only correlate with one DHCP server group. Using **dhcp relay server-select** repeatedly overwrites the previous configuration. However, if the specified DHCP server group does not exist, the interface still uses the previous correlation.

The *group-id* argument in **dhcp relay server-select** is configured by using **dhcp relay server-group**.

Configuring the DHCP relay agent security functions

Creating static bindings and enabling address check

To avoid invalid IP address configuration, configure address check on the DHCP relay agent.

With this feature enabled, the DHCP relay agent can dynamically record clients' IP-to-MAC bindings after the clients obtain IP addresses through DHCP. configure static IP-to-MAC bindings on the DHCP relay agent so that users can access external networks using fixed IP addresses.

Upon receiving a packet from a host, the DHCP relay agent checks the source IP and MAC addresses in the packet against the recorded dynamic and static bindings. If no match is found, the DHCP relay agent does not learn the ARP entry of the host, and does not forward any reply to the host, which thus cannot access external networks via the DHCP relay agent.

To create a static binding and enable address check:

| Step | Command | Remarks |
|-----------------------------|---|---|
| 1. Enter system view. | system-view | — |
| 2. Create a static binding. | dhcp relay security static ip-address mac-address [interface interface-type interface-number] | Optional. No static binding is created by default. |
| 3. Enter interface view. | interface interface-type interface-number | — |
| 4. Enable address check. | dhcp relay address-check enable | Required. Disabled by default. |

The **dhcp relay address-check enable** command can only be executed on Layer 3 Ethernet interfaces and VLAN interfaces.

Before enabling address check on an interface, you must enable the DHCP service, and enable the DHCP relay agent on the interface; otherwise, the address check configuration is ineffective.

The **dhcp relay address-check enable** command only checks IP and MAC addresses of clients.

When using **dhcp relay security static** to bind an interface to a static binding entry, make sure that the interface is configured as a DHCP relay agent; otherwise, address entry conflicts can occur.

Configuring periodic refresh of dynamic client entries

A DHCP client unicasts a DHCP-RELEASE message to the DHCP server when releasing its dynamically obtained IP address. The DHCP relay agent simply conveys the message to the DHCP server and does not remove the IP-to-MAC binding. To solve this problem, the periodic refresh of dynamic client entries feature is introduced.

With this feature, the DHCP relay agent uses the IP address of a client and the MAC address of the DHCP relay interface to periodically send a DHCP-REQUEST message to the DHCP server.

- If the server returns a DHCP-ACK message or does not return any message within a specified interval, the DHCP relay agent ages out the client entry with this IP address. When receiving the

DHCP-ACK message, the DHCP relay agent sends a DHCP-RELEASE message to release the IP address for saving IP addresses.

- If the server returns a DHCP-NAK message, the relay agent keeps the client entry.

To configure periodic refresh of dynamic client entries:

| Step | Command | Remarks |
|---|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enable periodic refresh of dynamic client entries. | dhcp relay security refresh enable | Optional. Enabled by default. |
| 3. Configure the refresh interval. | dhcp relay security tracker { <i>interval</i> auto } | Optional. Auto by default. (Auto interval is calculated by the relay agent according to the number of client entries.) |

Enabling unauthorized DHCP server detection

Unauthorized DHCP servers can assign wrong IP addresses to DHCP clients.

With unauthorized DHCP servers detection enabled, the DHCP relay agent checks whether a request contains Option 54 (Server Identifier Option). If yes, the DHCP relay agent records the IP address in the option, which is the IP address of the DHCP server that assigned an IP address to the DHCP client, and records the receiving interface. The administrator can use this information to check for unauthorized DHCP servers.

To enable unauthorized DHCP server detection:

| Step | Command | Remarks |
|---|---------------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable unauthorized DHCP server detection. | dhcp relay server-detect | Required. Disabled by default. |

The DHCP relay agent only logs a DHCP server once.

Enabling DHCP starvation attack protection

A DHCP starvation attack occurs when an attacker constantly sends forged DHCP requests using different MAC addresses in the chaddr field. This exhausts the IP address resources of the DHCP server are exhausted so legitimate DHCP clients cannot obtain IP addresses. The DHCP server can also fail to work because of exhaustion of system resources.

- To relieve a DHCP starvation attack that uses DHCP packets encapsulated with different source MAC addresses, limit the number of ARP entries that a Layer 3 interface can learn. also configure an interface that has learned the maximum MAC addresses to discard packets whose source MAC addresses are not in the MAC address table.
- To prevent a DHCP starvation attack that uses DHCP requests encapsulated with the same source MAC address, enable MAC address check on the DHCP relay agent. With this function enabled, the DHCP relay agent compares the chaddr field of a received DHCP request with the source MAC address field of the frame. If they are the same, the DHCP relay agent decides this request as valid and forwards it to the DHCP server; if not, the DHCP request is discarded.

To enable MAC address check:

| Step | Command | Remarks |
|------------------------------|--|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Enable MAC address check. | dhcp relay check mac-address | Required. Disabled by default. |

DHCP relay agents change the source MAC addresses when forwarding DHCP packets. Therefore, only enable MAC address check on a DHCP relay agent directly connected to DHCP clients. Otherwise, valid DHCP packets can be discarded and clients cannot obtain IP addresses.

Enabling offline detection

The DHCP relay agent checks whether a user is online by learning the ARP entry. When an ARP entry is aged out, the corresponding client is considered to be offline.

With this function enabled on an interface, the DHCP relay agent removes a client's IP-to-MAC entry when it is aged out, and sends a DHCP-RELEASE message to the DHCP server to release the IP address of the client.

To enable offline detection:

| Step | Command | Remarks |
|------------------------------|--|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Enable offline detection. | dhcp relay client-detect enable | Required. Disabled by default. |

Removing an ARP entry manually does not remove the corresponding client's IP-to-MAC binding. When the client goes offline, use the **undo dhcp relay security** command to remove the IP-to-MAC binding manually.

Configuring the DHCP relay agent to release an IP address

CAUTION:

- The DHCP relay agent can only generate client entries after you enable address check or IP Source Guard on the DHCP relay agent. For more information about IP source guard, see the *Security Configuration Guide*.
- A client's IP address to be released must be recorded by the relay agent. Otherwise, the DHCP relay agent cannot release the IP address.

After you configure the DHCP relay agent to release a client's IP address, the DHCP relay agent sends a DHCP-RELEASE message that contains the specified IP address to the server. Upon receiving the DHCP-RELEASE message, the DHCP server releases the IP address. The DHCP relay agent removes the client entry.

To configure the DHCP relay agent to release an IP address:

| Step | Command | Remarks |
|---|---|-----------|
| 1. Enter system view. | system-view | — |
| 2. Configure the DHCP relay agent to release an IP address. | dhcp relay release ip <i>client-ip</i> | Required. |

Configuring the DHCP relay agent to support Option 82

Configuration prerequisites

Before performing this configuration, complete the following tasks:

- Enable DHCP
- Enable the DHCP relay agent on the specified interface.
- Correlate a DHCP server group with relay agent interfaces.

Configuration procedure

To configure the DHCP relay agent to support Option 82:

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface <i>interface-type</i> <i>interface-number</i> | — |
| 3. Enable the relay agent to support Option 82. | dhcp relay information enable | Required. Disabled by default. |
| 4. Configure the handling strategy for requesting messages containing Option 82. | dhcp relay information strategy { drop keep replace } | Optional. Replace by default. |

| Step | | Command | Remarks |
|--|--|--|---|
| | Configure the padding format for Option 82. | dhcp relay information format { normal verbose [node-identifier { mac sysname user-defined node-identifier }] } | Optional. Normal by default. |
| 5. Configure non-user-defined Option 82. | Configure the code type for the circuit ID sub-option. | dhcp relay information circuit-id format-type { ascii hex } | Optional. By default, the code type depends on the padding format of Option 82. Each field has its own code type. The code type configuration only applies to non-user-defined Option 82. |
| | Configure the code type for the remote ID sub-option. | dhcp relay information remote-id format-type { ascii hex } | Optional. By default, the code type is hex . This code type configuration only applies to non-user-defined Option 82. |
| | Configure the padding content for the circuit ID sub-option. | dhcp relay information circuit-id string <i>circuit-id</i> | Optional. By default, the padding content depends on the padding format of Option 82. |
| 6. Configure user-defined Option 82. | Configure the padding content for the remote ID sub-option. | dhcp relay information remote-id string { <i>remote-id</i> sysname } | Optional. By default, the padding content depends on the padding format of Option 82. |

To support Option 82, perform related configuration on both the DHCP server and relay agent. See the chapter “DHCP server configuration” for DHCP server configuration of this kind.

If the handling strategy of the DHCP relay agent is configured as **replace**, you need to configure a padding format for Option 82. If the handling strategy is **keep** or **drop**, you need not configure any padding format.

If sub-option 1 (node identifier) of Option 82 is padded with the device name (sysname) of a node, the device name must contain no spaces. Otherwise, the DHCP relay agent drops the message.

Displaying and maintaining the DHCP relay agent

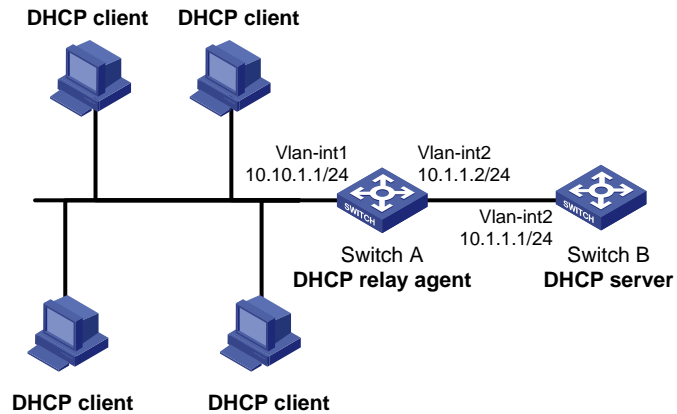
| Task | Command | Remarks |
|--|---|------------------------|
| Display information about DHCP server groups correlated to a specified or all interfaces. | display dhcp relay { all interface <i>interface-type interface-number</i> } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display Option 82 configuration information on the DHCP relay agent. | display dhcp relay information { all interface <i>interface-type interface-number</i> } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display information about bindings of DHCP relay agents. | display dhcp relay security [<i>ip-address</i> dynamic static] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display statistics information about bindings of DHCP relay agents. | display dhcp relay security statistics [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display information about the refreshing interval for entries of dynamic IP-to-MAC bindings. | display dhcp relay security tracker [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display information about the configuration of a specified or all DHCP server groups. | display dhcp relay server-group { <i>group-id</i> all } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display packet statistics on relay agent. | display dhcp relay statistics [server-group { <i>group-id</i> all }] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Clear packet statistics from relay agent. | reset dhcp relay statistics [server-group <i>group-id</i>] | Available in user view |

DHCP relay agent configuration examples

Network requirements

As shown in [Figure 30](#), DHCP clients reside on network 10.10.1.0/24. The IP address of the DHCP server is 10.1.1.1/24. Because the DHCP clients reside on a different network with the DHCP server, a DHCP relay agent is deployed to forward messages between DHCP clients and the DHCP server. VLAN-interface 1 on the DHCP relay agent (Switch A) connects to the network where DHCP clients reside. The IP address of VLAN-interface 1 is 10.10.1.1/24 and the IP address of VLAN-interface 2 is 10.1.1.2/24.

Figure 30 Network diagram for DHCP relay agent



Configuration procedure

Specify IP addresses for the interfaces (omitted).

Enable DHCP.

```
<SwitchA> system-view
[SwitchA] dhcp enable
```

Add DHCP server 10.1.1.1 into DHCP server group 1.

```
[SwitchA] dhcp relay server-group 1 ip 10.1.1.1
```

Enable the DHCP relay agent on VLAN-interface 1.

```
[SwitchA] interface vlan-interface 1
[SwitchA-Vlan-interface1] dhcp select relay
```

Correlate VLAN-interface 1 to DHCP server group 1.

```
[SwitchA-Vlan-interface1] dhcp relay server-select 1
```

After the preceding configuration is complete, DHCP clients can obtain IP addresses and other network parameters through the DHCP relay agent from the DHCP server. Use **display dhcp relay statistics** to view statistics of DHCP packets forwarded by DHCP relay agents. After you enable address check of the DHCP relay agents with **dhcp relay address-check enable**, use **display dhcp relay security** to view bindings of DHCP relay agents.

Because the DHCP relay agent and server are on different subnets, you must configure a static route or dynamic routing protocol to make them reachable to each other.

Configurations on the DHCP server are also required to guarantee the client-server communication via the DHCP relay agent. For DHCP server configuration information, see the chapter “DHCP server configuration.”

DHCP relay agent Option 82 support configuration example

Network requirements

- As shown in Figure 30, Enable Option 82 on the DHCP relay agent (Switch A).
- Configure the handling strategy for DHCP requests containing Option 82 as **replace**.
- Configure the padding content for the circuit ID sub-option as **company001** and for the remote ID sub-option as **device001**.

- Switch A forwards DHCP requests to the DHCP server (Switch B) after replacing Option 82 in the requests, so that the DHCP clients can obtain IP addresses.

Configuration procedure

```
# Specify IP addresses for the interfaces (omitted).

# Enable DHCP.
<SwitchA> system-view
[SwitchA] dhcp enable

# Add DHCP server 10.1.1.1 into DHCP server group 1.
[SwitchA] dhcp relay server-group 1 ip 10.1.1.1

# Enable the DHCP relay agent on VLAN-interface 1.
[SwitchA] interface vlan-interface 1
[SwitchA-Vlan-interface1] dhcp select relay

# Correlate VLAN-interface 1 to DHCP server group 1.
[SwitchA-Vlan-interface1] dhcp relay server-select 1

# Enable the DHCP relay agent to support Option 82, and perform Option 82-related configurations.
[SwitchA-Vlan-interface1] dhcp relay information enable
[SwitchA-Vlan-interface1] dhcp relay information strategy replace
[SwitchA-Vlan-interface1] dhcp relay information circuit-id string company001
[SwitchA-Vlan-interface1] dhcp relay information remote-id string device001
```

Configurations on the DHCP server are also required to make the Option 82 configurations function normally.

Troubleshooting DHCP relay agent configuration

Symptom

DHCP clients cannot obtain any configuration parameters via the DHCP relay agent.

Analysis

Some problems can occur with the DHCP relay agent or server configuration. Enable debugging and execute **display** on the DHCP relay agent to view the debugging information and interface state information for locating the problem.

Solution

To locate the problem, enable debugging and execute **display** on the DHCP relay agent to view the debugging information and interface state information.

Check that:

- The DHCP is enabled on the DHCP server and relay agent.
- The address pool on the same subnet where DHCP clients reside is available on the DHCP server.
- The routes between the DHCP server and DHCP relay agent are reachable.
- The relay agent interface connected to DHCP clients is correlated with correct DHCP server group and IP addresses for the group members are correct.

DHCP client configuration

The DHCP client configuration is only supported on Layer 3 Ethernet interfaces, and VLAN interfaces.

When multiple VLAN interfaces with the same MAC address use DHCP for IP address acquisition via a relay agent, the DHCP server cannot be a Windows 2000 Server or Windows 2003 Server.

With the DHCP client enabled, an interface uses DHCP to obtain configuration parameters such as an IP address from the DHCP server.

Enabling the DHCP client on an interface

| Step | Command | Remarks |
|---|---|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Enable the DHCP client on the interface. | ip address dhcp-alloc [client-identifier mac interface-type interface-number] | Required. Disabled by default. |

An interface can be configured to acquire an IP address in multiple ways, but these ways are mutually exclusive. The latest configuration overwrites the previous one.

After the DHCP client is enabled on an interface, no secondary IP address can be configured for the interface.

If the IP address that interface A obtains from the DHCP server is on the same subnet as the IP address of interface B, interface A neither uses the IP address nor requests any IP address from the DHCP server, unless the IP address of interface B is manually deleted and interface A is brought up again by first executing **shutdown** and then **undo shutdown** or the DHCP client is re-enabled on interface A by executing **undo ip address dhcp-alloc** and then **ip address dhcp-alloc**.

Displaying and maintaining the DHCP client

| Task | Command | Remarks |
|--|--|-----------------------|
| Display specified configuration information. | display dhcp client [verbose] [interface interface-type interface-number] [{ begin exclude include } regular-expression] | Available in any view |

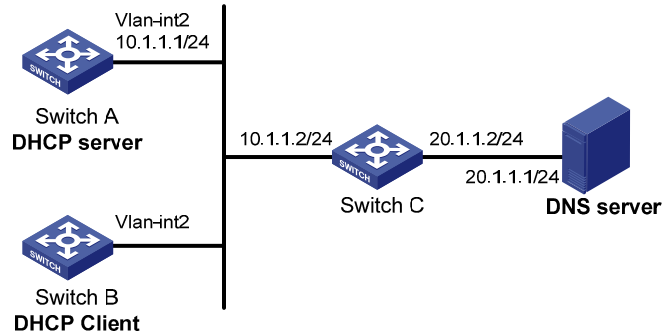
DHCP client configuration example

Network requirements

As shown in [Figure 31](#), on a LAN, Switch B contacts the DHCP server via VLAN-interface 2 to obtain an IP address, DNS server address, and static route information. The IP address resides on network 10.1.1.0/24. The DNS server address is 20.1.1.1. The next hop of the static route to network 20.1.1.0/24 is 10.1.1.2.

The DHCP server uses Option 121 to assign static route information to DHCP clients. The destination descriptor field comprises two parts, subnet mask length and destination network address. In this example, the value of the destination descriptor field takes 18 14 01 01, a hexadecimal number indicating that the subnet mask length is 24 and destination network address is 20.1.1.0, and the value of the next hop address field takes 0A 01 01 02, a hexadecimal number indicating that the next hop is 10.1.1.2.

Figure 31 Network diagram for DHCP client configuration example



Configuration procedure

1. Configure Switch A

Specify the IP address of VLAN-interface 2.

```

<SwitchA> system-view
[SwitchA] interface vlan-interface 2
[SwitchA-Vlan-interface2] ip address 10.1.1.1 24
[SwitchA-Vlan-interface2] quit
  
```

Enable the DHCP service.

```
[SwitchA] dhcp enable
```

Exclude an IP address from automatic allocation.

```
[SwitchA] dhcp server forbidden-ip 10.1.1.2
```

Configure DHCP address pool 0 and specify the subnet, lease duration, DNS server address, and a static route to subnet 20.1.1.0/24.

```

[SwitchA] dhcp server ip-pool 0
[SwitchA-dhcp-pool-0] network 10.1.1.0 mask 255.255.255.0
[SwitchA-dhcp-pool-0] expired day 10
[SwitchA-dhcp-pool-0] dns-list 20.1.1.1
[SwitchA-dhcp-pool-0] option 121 hex 18 14 01 01 0A 01 01 02
  
```

2. Configure Switch B

Enable the DHCP client on VLAN-interface 2.

```

<SwitchB> system-view
[SwitchB] interface vlan-interface 2
[SwitchB-Vlan-interface2] ip address dhcp-alloc
  
```

3. Verification

Use **display dhcp client** to view the IP address and other network parameters assigned to Switch B.

```
[SwitchB-Vlan-interface2] display dhcp client verbose
Vlan-interface2 DHCP client information:
Current machine state: BOUND
Allocated IP: 10.1.1.3 255.255.255.0
Allocated lease: 864000 seconds, T1: 432000 seconds, T2: 756000 seconds
Lease from 2009.02.20 11:06:35 to 2009.03.02 11:06:35
DHCP server: 10.1.1.1
Transaction ID: 0x410090f0
Classless static route:
  Destination: 20.1.1.0, Mask: 255.255.255.0, NextHop: 10.1.1.2
DNS server: 20.1.1.1
Client ID: 3030-3066-2e65-3230-
          302e-3030-3032-2d45-
          7468-6572-6e65-7430-
          2f30
T1 will timeout in 4 days 23 hours 59 minutes 50 seconds.
```

Use **display ip routing-table** to view the route information on Switch B. A static route to network 20.1.1.0/24 is added to the routing table.

```
[SwitchB-Vlan-interface2] display ip routing-table
Routing Tables: Public
          Destinations : 5          Routes : 5

Destination/Mask    Proto  Pre  Cost           NextHop         Interface
-----
10.1.1.0/24         Direct  0    0             10.1.1.3        Vlan2
10.1.1.3/32         Direct  0    0             127.0.0.1       InLoop0
20.1.1.0/24         Static  70   0             10.1.1.2        Vlan2
127.0.0.0/8         Direct  0    0             127.0.0.1       InLoop0
127.0.0.1/32       Direct  0    0             127.0.0.1       InLoop0
```

DHCP snooping configuration

The DHCP snooping-enabled device must be either between the DHCP client and relay agent, or between the DHCP client and server. It does not work if it is between the DHCP relay agent and DHCP server.

Snooping functions

DHCP snooping can implement the following functions:

- Ensure DHCP clients to obtain IP addresses from authorized DHCP servers
- Record IP-to-MAC mappings of DHCP clients

Ensuring DHCP clients to obtain IP addresses from authorized DHCP servers

With DHCP snooping, the ports of a switch can be configured as trusted or untrusted to make sure that clients only obtains IP addresses from authorized DHCP servers.

- **Trusted**—A trusted port forwards DHCP messages normally to ensure the clients get IP addresses from an authorized DHCP server.
- **Untrusted**—An untrusted port discards received DHCP-ACK and DHCP-OFFER messages to avoid IP address allocation from any unauthorized server.

Configure ports that connect to authorized DHCP servers or other DHCP snooping devices as trusted, and other ports as untrusted.

Recording DHCP client IP-to-MAC mappings

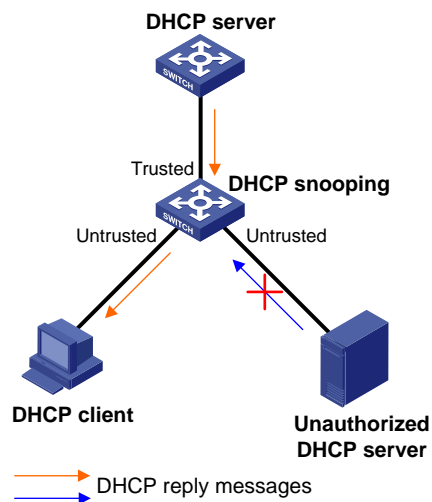
DHCP snooping reads DHCP-REQUEST messages and DHCP-ACK messages from trusted ports to record DHCP snooping entries. A DHCP snooping entry includes the MAC and IP addresses of the clients, the port that connects to DHCP clients, and the VLAN of the port. With DHCP snooping entries, DHCP snooping can implement the following functions:

- **ARP detection**—Whether ARP packets are sent from an authorized client is determined based on DHCP snooping entries. This feature prevents ARP attacks from unauthorized clients. For more information, see the *Security Configuration Guide*.
- **MFF**—In automatic mode, after receiving an ARP request from a client, the MFF device searches DHCP snooping entries for the corresponding gateway address, and sends the gateway MAC address to the client. This feature allows the gateway to monitor client traffic, prevent malicious attacks among clients, and therefore guarantees network security. For more information, see *Security Configuration Guide*.
- **IP Source Guard**—IP Source Guard uses dynamic binding entries generated by DHCP snooping to filter packets on a per-port basis, and prevents unauthorized packets from traveling through. For more information, see *Security Configuration Guide*.
- **VLAN mapping**—The switch replaces SVLANs in packets with CVLANs by searching corresponding DHCP snooping entries for DHCP client information including IP addresses, MAC addresses, and CVLANs, before sending the packets to clients. For more information, see *Layer 2—LAN Switching*.

Application environment of trusted ports

Configuring a trusted port connected to a DHCP server

Figure 32 Configure trusted and untrusted ports



As shown in Figure 32, the trusted port forwards reply messages from the DHCP server to the client, but the untrusted port connected to the unauthorized DHCP server cannot forward any reply messages. This ensures that the DHCP client can obtain an IP address from the authorized DHCP server.

Configuring trusted ports in a cascaded network

In a cascaded network involving multiple DHCP snooping devices, the ports connected to other DHCP snooping devices should be configured as trusted ports.

To save system resources, disable the trusted ports, which are indirectly connected to DHCP clients, from recording client IP-to-MAC bindings upon receiving DHCP requests.

Figure 33 Configure trusted ports in a cascaded network

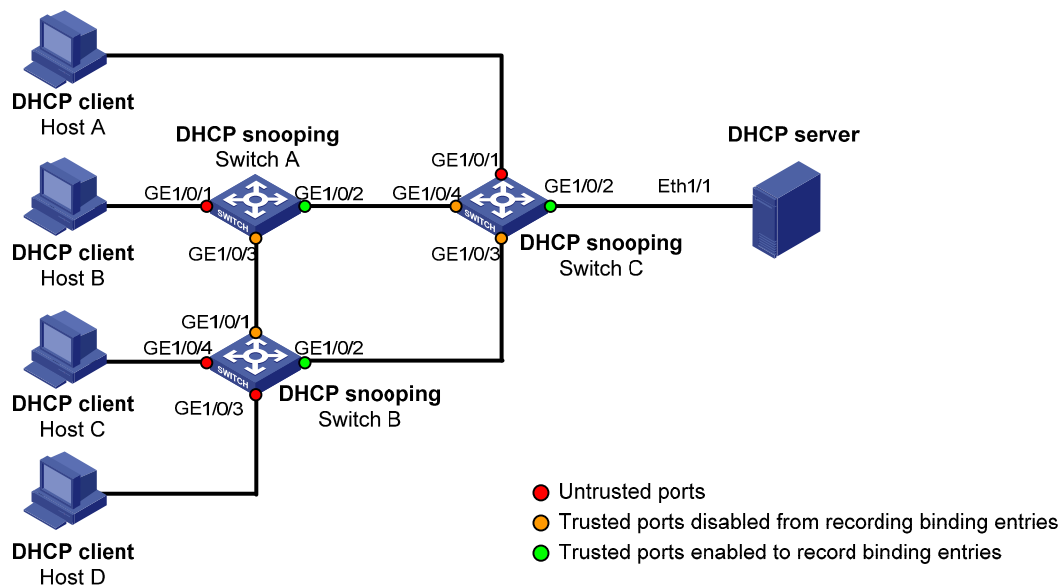


Table 4 describes roles of the ports shown in Figure 33.

Table 4 Roles of ports

| Device | Untrusted port | Trusted port disabled from recording binding entries | Trusted port enabled to record binding entries |
|---------------|---|---|---|
| Switch A | GigabitEthernet 1/0/1 | GigabitEthernet 1/0/3 | GigabitEthernet 1/0/2 |
| Switch B | GigabitEthernet 1/0/3 and GigabitEthernet 1/0/4 | GigabitEthernet 1/0/1 | GigabitEthernet 1/0/2 |
| Switch C | GigabitEthernet 1/0/1 | GigabitEthernet 1/0/3 and GigabitEthernet 1/0/4 | GigabitEthernet 1/0/2 |

DHCP snooping support for Option 82

Option 82 records the location information of the DHCP client, so the administrator can locate the DHCP client for security control and accounting purposes. For more information, see the chapter “DHCP relay agent configuration.”

If DHCP snooping supports Option 82, it handles a client’s request according to the contents defined in Option 82, if any. The handling strategies are described in Table 5.

If a reply returned by the DHCP server contains Option 82, the DHCP snooping device removes the Option 82 before forwarding the reply to the client. If the reply contains no Option 82, the DHCP snooping device forwards it directly.

Table 5 Handling strategies of DHCP snooping

| If a client's requesting message has... | Handling strategy | Padding format | The DHCP snooping device will... |
|---|-------------------|---|---|
| Option 82 | Drop | — | Drop the message. |
| | Replace | Random | Forward the message without changing Option 82. |
| | | Normal | Forward the message after replacing the original Option 82 with the Option 82 padded in normal format. |
| | | Verbose | Forward the message after replacing the original Option 82 with the Option 82 padded in verbose format. |
| | User-defined | Forward the message after replacing the original Option 82 with the user-defined Option 82. | |
| No Option 82 | — | Normal | Forward the message after adding the Option 82 padded in normal format. |
| | — | Verbose | Forward the message after adding the Option 82 padded in verbose format. |
| | — | User-defined | Forward the message after adding the user-defined Option 82. |

The handling strategy and padding format for Option 82 on the DHCP snooping device are the same as those on the relay agent.

Configuration task list

| Task | Remarks |
|---|-----------|
| Configuring DHCP snooping basic functions | Required. |
| Configuring DHCP snooping to support Option 82 | Optional. |
| Configuring DHCP snooping entries backup | Optional. |
| Enabling DHCP starvation attack protection | Optional. |
| Enabling DHCP-REQUEST message attack protection | Optional. |
| Configuring DHCP packet rate limit | Optional. |

Configuring DHCP snooping basic functions

| Step | Command | Remarks |
|---|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enable DHCP snooping. | dhcp-snooping | Required. Disabled by default. |
| 3. Enter interface view. | interface interface-type interface-number | — |
| 4. Specify the port as a trusted port that records the IP-to-MAC bindings of clients. | dhcp-snooping trust | Required. After DHCP snooping is enabled, a port is an untrusted port by default. |
| 5. Return to system view. | quit | — |
| 6. Enter interface view. | interface interface-type interface-number | — The interface indirectly connects to the DHCP client. |
| 7. Specify the port as a trusted port that does not record the IP-to-MAC bindings of clients. | dhcp-snooping trust no-user-binding | Optional. After DHCP snooping is enabled, a port is an untrusted port by default. |

You must specify the ports connected to the authorized DHCP servers as trusted to make sure that DHCP clients can obtain valid IP addresses. The trusted port and the port connected to the DHCP client must be in the same VLAN.

Specify Layer 2 Ethernet interfaces and Layer 2 aggregate interfaces as trusted ports. For more information about aggregate interfaces, see *Layer 2—LAN Switching Configuration Guide*.

If a Layer 2 Ethernet interface is added to an aggregation group, the DHCP snooping configuration of the interface does not take effect. After the interface quits the aggregation group, the configuration is effective.

DHCP snooping can work with basic QinQ or flexible QinQ. When receiving a packet without any VLAN tag from the DHCP client to the DHCP server, the DHCP snooping device adds a VLAN tag to the packet. If the packet has one VLAN tag, the device adds another VLAN tag to the packet and records the two VLAN tags in a DHCP snooping entry. The newly added VLAN tag is the outer tag. If the packet has two VLAN tags, the device directly forwards the packet to the DHCP server without adding any tag. If you need to add a new VLAN tag and meanwhile modify the original VLAN tag for the packet, DHCP snooping cannot work with flexible QinQ.

Configuring DHCP snooping to support Option 82

| Step | Command | Remarks | |
|--|---|--|--|
| 1. Enter system view. | system-view | — | |
| 2. Enter interface view. | interface interface-type interface-number | — | |
| 3. Enable DHCP snooping to support Option 82. | dhcp-snooping information enable | Required. Disabled by default. | |
| 4. Configure the handling strategy for requesting messages containing Option 82. | dhcp-snooping information strategy { drop keep replace } | Optional. Replace by default. | |
| | Configure the padding format for Option 82. | dhcp-snooping information format { normal verbose [node-identifier { mac sysname user-defined node-identifier }] } | Optional. Normal by default. |
| 5. Configure non-user-defined Option 82. | Configure the code type for the circuit ID sub-option. | dhcp-snooping information circuit-id format-type { ascii hex } | Optional. By default, the code type depends on the padding format of Option 82. Each field has its own code type. This code type configuration only applies to non-user-defined Option 82. |
| | Configure the code type for the remote ID sub-option. | dhcp-snooping information remote-id format-type { ascii hex } | Optional. Hex by default. The code type configuration only applies to non-user-defined Option 82. |
| 6. Configure user-defined Option 82. | Configure the padding content for the circuit ID sub-option. | dhcp-snooping information [vlan <i>vlan-id</i>] circuit-id string <i>circuit-id</i> | Optional. By default, the padding content depends on the padding format of Option 82. |
| | Configure the padding content for the remote ID sub-option. | dhcp-snooping information [vlan <i>vlan-id</i>] remote-id string { <i>remote-id</i> sysname } | Optional. By default, the padding content depends on the padding format of Option 82. |

Enable DHCP snooping to support only Option 82 on Layer 2 Ethernet interfaces and Layer 2 aggregate interfaces.

If a Layer 2 Ethernet interface is added to an aggregation group, enabling DHCP snooping to support Option 82 on the interface does not take effect. After the interface quits the aggregation group, the configuration is effective.

To support Option 82, perform related configuration on both the DHCP server and the switch enabled with DHCP snooping. See the chapter “DHCP server configuration” for DHCP server configuration of this kind.

If the handling strategy of the DHCP-snooping-enabled device is configured as **replace**, you need to configure a padding format for Option 82. If the handling strategy is **keep** or **drop**, you need not configure any padding format.

If the Option 82 is padded with the device name (sysname) of a node, the device name must contain no spaces. Otherwise, the DHCP-snooping-enabled device drops the message. Use **sysname** to specify the device name. For more information about this command, see Device management commands in the *Fundamentals Command Reference*.

If DHCP snooping and QinQ work together or the DHCP snooping device receives a DHCP packet with two VLAN tags, and the normal or verbose padding format is adopted for Option 82, DHCP snooping fills the VLAN ID field of sub-option 1 with outer VLAN tag.inter VLAN tag. For example, if the outer VLAN tag is 10 (a in hexadecimal) and the inner VLAN tag is 20 (14 in hexadecimal), the VLAN ID is 000a.0014.

Configuring DHCP snooping entries backup

DHCP snooping entries cannot survive a reboot. If the DHCP snooping device is rebooted, security modules (such as IP source guard) that use DHCP snooping entries to authenticate users rejects requests from clients until new entries are learned.

The DHCP snooping entries backup feature enables you to store DHCP snooping entries in a file. When the DHCP snooping device reboots, it reads DHCP snooping entries from this file.

To configure DHCP snooping entries backup:

| Step | Command | Remarks |
|---|---|---|
| 1. Enter system view. | system-view | — |
| 2. Specify the name of the file for storing DHCP snooping entries. | dhcp-snooping binding database filename filename | Required. Not specified by default. DHCP snooping entries are stored immediately after this command is used and then updated at the interval set by dhcp-snooping binding database update interval . |
| 3. Back up DHCP snooping entries to the file. | dhcp-snooping binding database update now | Optional. DHCP snooping entries is stored to the file each time this command is used. |
| 4. Set the interval at which the DHCP snooping entry file is refreshed. | dhcp-snooping binding database update interval minutes | Optional. By default, the file is not refreshed periodically. |

After DHCP snooping is disabled with **undo dhcp-snooping**, the switch deletes all DHCP snooping bindings, including those stored in the file.

Enabling DHCP starvation attack protection

A DHCP starvation attack occurs when an attacker constantly sends forged DHCP requests using different MAC addresses in the chaddr field to a DHCP server. This exhausts the IP address resources of the DHCP server so legitimate DHCP clients cannot obtain IP addresses. The DHCP server can also fail to work because of exhaustion of system resources.

- To relieve a DHCP starvation attack that uses DHCP packets encapsulated with different source MAC addresses, limit the number of MAC addresses that a Layer 2 port can learn.
- To prevent a DHCP starvation attack that uses DHCP requests encapsulated with the same source MAC address, enable MAC address check on the DHCP snooping device. With this function enabled, the DHCP snooping device compares the chaddr field of a received DHCP request with the source MAC address field of the frame. If they are the same, the request is considered valid and forwarded to the DHCP server; if not, the request is discarded.

To enable MAC address check:

| Step | Command | Remarks |
|------------------------------|--|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Enable MAC address check. | dhcp-snooping check mac-address | Required. Disabled by default. |

Enable MAC address check only on Layer 2 Ethernet interfaces and Layer 2 aggregate interfaces.

Enabling DHCP-REQUEST message attack protection

Attackers can forge DHCP-REQUEST messages to renew the IP address leases for legitimate DHCP clients that no longer need the IP addresses. These forged messages keep a victim DHCP server renewing the leases of IP addresses instead of releasing the IP addresses. This wastes IP address resources.

To prevent such attacks, enable DHCP-REQUEST message check on DHCP snooping devices. With this feature enabled, upon receiving a DHCP-REQUEST message, a DHCP snooping device looks up local DHCP snooping entries for the corresponding entry of the message. If an entry is found, the DHCP snooping device compares the entry with the message information. If they are consistent, the DHCP-REQUEST message is considered as a valid lease renewal request and forwarded to the DHCP server. If they are not consistent, the message is considered as a forged lease renewal request and discarded. If no corresponding entry is found locally, the message is considered valid and forwarded to the DHCP server.

To enable DHCP-REQUEST message check:

| Step | Command | Remarks |
|---------------------------------------|--|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Enable DHCP-REQUEST message check. | dhcp-snooping check request-message | Required. Disabled by default. |

Enable DHCP-REQUEST message check only on Layer 2 Ethernet interfaces and Layer 2 aggregate interfaces.

Configuring DHCP packet rate limit

To identify DHCP packets from unauthorized DHCP servers, DHCP snooping delivers all incoming DHCP packets to the CPU. If a malicious user sends a large number of DHCP requests to the DHCP snooping device, the CPU of the device is overloaded, and the device can even crash. To solve this problem, configure DHCP packet rate limit on relevant interfaces.

To configure DHCP packet rate limit:

| Step | Command | Remarks |
|---|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter Layer 2 Ethernet interface view or Layer 2 aggregate interface view. | interface interface-type interface-number | — |
| 3. Configure the maximum rate of incoming DHCP packets. | dhcp-snooping rate-limit rate | Required. Not configured by default. |

Configure DHCP packet rate limit only on Layer 2 Ethernet interfaces and Layer 2 aggregate interfaces.

If a Layer 2 Ethernet interface belongs to an aggregation group, it uses the DHCP packet maximum rate configured on the corresponding Layer 2 aggregate interface.

Displaying and maintaining DHCP snooping

| Task | Command | Remarks |
|--|--|------------------------|
| Display DHCP snooping entries. | display dhcp-snooping [ip ip-address] [{ begin exclude include } regular-expression] | Available in any view |
| Display Option 82 configuration information on the DHCP snooping device. | display dhcp-snooping information { all interface interface-type interface-number } [{ begin exclude include } regular-expression] | Available in any view |
| Display DHCP packet statistics on the DHCP snooping device. | display dhcp-snooping packet statistics [slot slot-number] [{ begin exclude include } regular-expression] | Available in any view |
| Display information about trusted ports. | display dhcp-snooping trust [{ begin exclude include } regular-expression] | Available in any view |
| Display the DHCP snooping entry file information. | display dhcp-snooping binding database [{ begin exclude include } regular-expression] | Available in any view |
| Clear DHCP snooping entries. | reset dhcp-snooping { all ip ip-address } | Available in user view |
| Clear DHCP packet statistics on the DHCP snooping device. | reset dhcp-snooping packet statistics [slot slot-number] | Available in user view |

Configuration examples

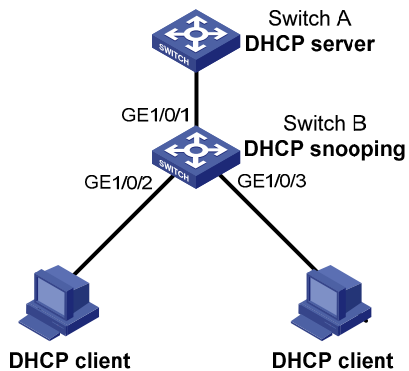
DHCP snooping configuration example

Network requirements

As shown in [Figure 34](#), Switch B is connected to a DHCP server through GigabitEthernet 1/0/1, and to two DHCP clients through GigabitEthernet 1/0/2 and GigabitEthernet 1/0/3. GigabitEthernet 1/0/1 forwards DHCP server responses while the other two do not.

Switch B records clients' IP-to-MAC address bindings in DHCP-REQUEST messages and DHCP-ACK messages received from trusted ports.

Figure 34 Network diagram for DHCP snooping configuration



Configuration procedure

Enable DHCP snooping.

```
<SwitchB> system-view  
[SwitchB] dhcp-snooping
```

Specify GigabitEthernet 1/0/1 as trusted.

```
[SwitchB] interface GigabitEthernet 1/0/1  
[SwitchB-GigabitEthernet1/0/1] dhcp-snooping trust  
[SwitchB-GigabitEthernet1/0/1] quit
```


DHCP snooping Option 82 support configuration example

Network requirements

- As shown in [Figure 34](#), enable DHCP snooping and Option 82 support on Switch B.
- Configure the handling strategy for DHCP requests containing Option 82 as **replace**.
- On GigabitEthernet 1/0/2, configure the padding content for the circuit ID sub-option as **company001** and for the remote ID sub-option as **device001**.
- On GigabitEthernet 1/0/3, configure the padding format as **verbose**, access node identifier as **sysname**, and code type as **ascii** for Option 82.
- Switch B forwards DHCP requests to the DHCP server (Switch A) after replacing Option 82 in the requests, so that the DHCP clients can obtain IP addresses.

Configuration procedure

Enable DHCP snooping.

```
<SwitchB> system-view  
[SwitchB] dhcp-snooping
```

Specify GigabitEthernet 1/0/1 as trusted.

```
[SwitchB] interface GigabitEthernet 1/0/1  
[SwitchB-GigabitEthernet1/0/1] dhcp-snooping trust  
[SwitchB-GigabitEthernet1/0/1] quit
```

Configure GigabitEthernet 1/0/2 to support Option 82.

```
[SwitchB] interface GigabitEthernet 1/0/2  
[SwitchB-GigabitEthernet1/0/2] dhcp-snooping information enable  
[SwitchB-GigabitEthernet1/0/2] dhcp-snooping information strategy replace  
[SwitchB-GigabitEthernet1/0/2] dhcp-snooping information circuit-id string company001  
[SwitchB-GigabitEthernet1/0/2] dhcp-snooping information remote-id string device001  
[SwitchB-GigabitEthernet1/0/2] quit
```

Configure GigabitEthernet 1/0/3 to support Option 82.

```
[SwitchB] interface GigabitEthernet 1/0/3  
[SwitchB-GigabitEthernet1/0/3] dhcp-snooping information enable  
[SwitchB-GigabitEthernet1/0/3] dhcp-snooping information strategy replace  
[SwitchB-GigabitEthernet1/0/3] dhcp-snooping information format verbose node-identifier  
sysname  
[SwitchB-GigabitEthernet1/0/3] dhcp-snooping information circuit-id format-type ascii  
[SwitchB-GigabitEthernet1/0/3] dhcp-snooping information remote-id format-type ascii
```

BOOTP client configuration

BOOTP client configuration only applies to Layer 3 Ethernet interfaces, and VLAN interfaces.

If several VLAN interfaces sharing the same MAC address obtain IP addresses through a BOOTP relay agent, the BOOTP server cannot be a Windows 2000 Server or Windows 2003 Server.

Application

After you specify an interface of switch as a BOOTP client, the interface can use BOOTP to get information (such as IP address) from the BOOTP server, which simplifies your configuration.

To use BOOTP, an administrator must configure a BOOTP parameter file for each BOOTP client on the BOOTP server. The parameter file contains information such as MAC address and IP address of a BOOTP client. When a BOOTP client sends a request to the BOOTP server, the BOOTP server searches for the BOOTP parameter file and returns the corresponding configuration information.

BOOTP is usually used in relatively stable environments. In network environments change frequently, DHCP is more suitable.

Because a DHCP server can interact with a BOOTP client, use the DHCP server to configure an IP address for the BOOTP client, without any BOOTP server.

Dynamically obtaining an IP address

A DHCP server can take the place of the BOOTP server in the following dynamic IP address acquisition.

A BOOTP client dynamically obtains an IP address from a BOOTP server in the following steps:

1. The BOOTP client broadcasts a BOOTP request, which contains its own MAC address.
2. The BOOTP server receives the request and searches the configuration file for the corresponding IP address and other information according to the MAC address of the BOOTP client. The BOOTP server then returns a BOOTP response to the BOOTP client.
3. The BOOTP client obtains the IP address from the received response.

Protocols and standards

Some protocols and standards related to BOOTP include:

- RFC 951, *Bootstrap Protocol (BOOTP)*
- RFC 2132, *DHCP Options and BOOTP Vendor Extensions*
- RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*

Configuring an interface to dynamically obtain an IP address through BOOTP

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure an interface to dynamically obtain an IP address through BOOTP. | ip address bootp-alloc | Required. By default, an interface does not use BOOTP to obtain an IP address. |

Displaying and maintaining BOOTP client configuration

| Task | Command | Remarks |
|-----------------------------------|--|-----------------------|
| Display BOOTP client information. | display bootp client [interface interface-type interface-number] [{ begin exclude include } regular-expression] | Available in any view |

BOOTP client configuration example

Network requirements

As shown in [Figure 26](#), Switch B's port belonging to VLAN 1 is connected to the LAN. VLAN-interface 1 obtains an IP address from the DHCP server by using BOOTP.

Configuration procedure

The following only describes the configuration on Switch B serving as a client.

Configure VLAN-interface 1 to dynamically obtain an IP address from the DHCP server.

```
<SwitchB> system-view  
[SwitchB] interface vlan-interface 1  
[SwitchB-Vlan-interface1] ip address bootp-alloc
```

Use **display bootp client** to view the IP address assigned to the BOOTP client.

To make the BOOTP client obtain an IP address from the DHCP server, you must perform additional configurations on the DHCP server. For more information, see the chapter "DHCP server configuration."

IPv4 DNS configuration

DNS is a distributed database used by TCP/IP applications to translate domain names into corresponding IP addresses. With DNS, use easy-to-remember domain names in some applications and let the DNS server translate them into correct IP addresses.

DNS services can be static and dynamic. After a user specifies a name, the switch checks the local static name resolution table for an IP address. If no IP address is available, it contacts the DNS server for dynamic name resolution, which takes more time than static name resolution. Therefore, some frequently queried name-to-IP address mappings are stored in the local static name resolution table to improve efficiency.

Static domain name resolution

The static domain name resolution means setting up mappings between domain names and IP addresses. IP addresses of the corresponding domain names can be found in the static domain resolution table when you use applications such as Telnet.

Dynamic domain name resolution

Resolution process

1. A user program sends a name query to the resolver of the DNS client.
2. The DNS resolver looks up the local domain name cache for a match. If a match is found, it sends the corresponding IP address back. If not, it sends a query to the DNS server.
3. The DNS server looks up the corresponding IP address of the domain name in its DNS database. If no match is found, the server sends a query to a higher level DNS server. This process continues until a result, whether successful or not, is returned.
4. After receiving a response from the DNS server, the DNS client returns the resolution result to the application.

Figure 35 Dynamic domain name resolution

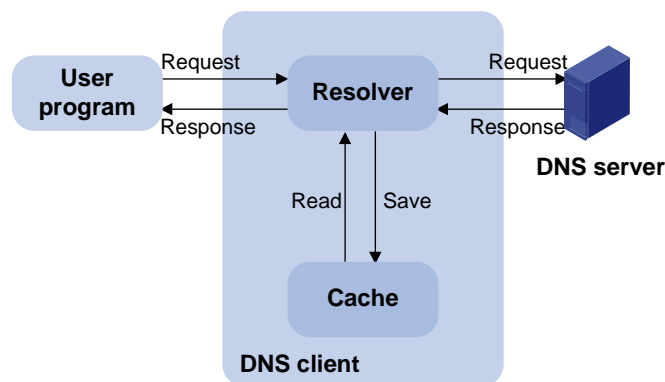


Figure 35 shows the relationship between the user program, DNS client, and DNS server.

The DNS client is made up of the resolver and cache. The user program and DNS client can run on the same device or different devices, but the DNS server and the DNS client usually run on different devices.

Dynamic domain name resolution allows the DNS client to store latest mappings between domain names and IP addresses in the dynamic domain name cache. There is no need to send a request to the DNS server for a repeated query next time. The aged mappings are removed from the cache after some time, and latest entries are required from the DNS server. The DNS server decides how long a mapping is valid, and the DNS client gets the aging information from DNS messages.

DNS suffixes

The DNS client holds a list of suffixes, which can be defined by users. It is used when the name to be resolved is incomplete. The resolver can supply the missing part.

For example, a user can configure com as the suffix for aabbcc.com. The user only needs to type aabbcc to obtain the IP address of aabbcc.com, because the resolver adds the suffix and delimiter before passing the name to the DNS server.

- If there is no dot in the domain name (for example, aabbcc), the resolver considers this a host name and adds a DNS suffix before query. If no match is found after all configured suffixes are used respectively, the original domain name (for example, aabbcc) is used for query.
- If there is a dot in the domain name (for example, www.aabbcc), the resolver directly uses this domain name for query. If the query fails, the resolver adds a DNS suffix for another query.
- If the dot is at the end of the domain name (for example, aabbcc.com.), the resolver considers it a FQDN and returns the query result, successful or failed. The dot (.) at the end of the domain name is considered a terminating symbol.

The switch supports static and dynamic DNS client services.

If an alias is configured for a domain name on the DNS server, the switch can resolve the alias into the IP address of the host.

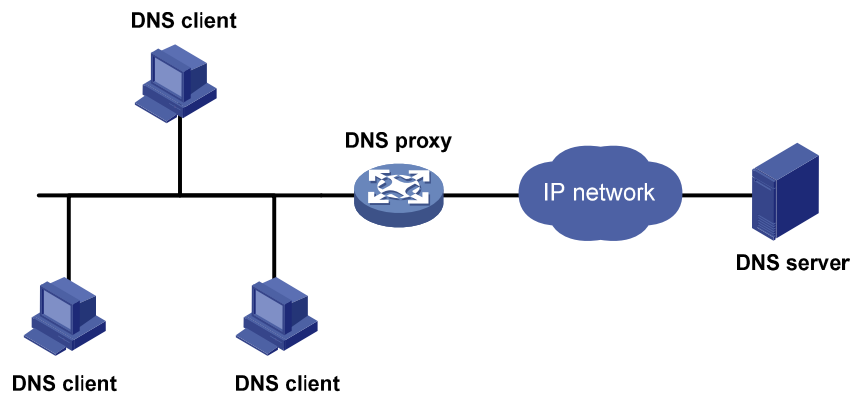
DNS proxy

A DNS proxy forwards DNS requests and replies between DNS clients and a DNS server.

As shown in [Figure 36](#), a DNS client sends a DNS request to the DNS proxy, which forwards the request to the designated DNS server, and conveys the reply from the DNS server to the client.

The DNS proxy simplifies network management. When the DNS server address is changed, change the configuration on only the DNS proxy instead of on each DNS client.

Figure 36 DNS proxy networking application



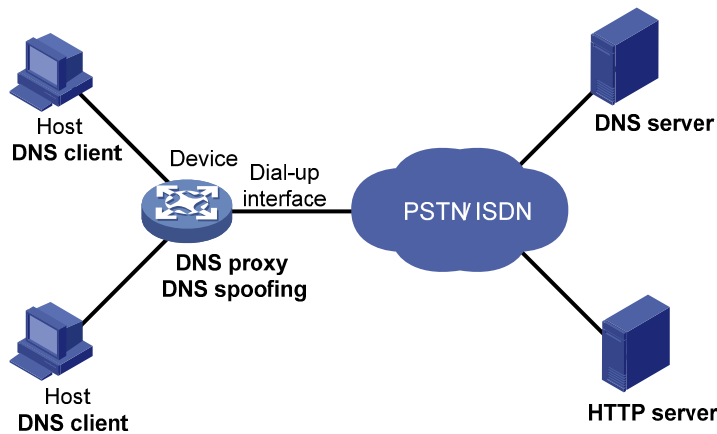
Operation of a DNS proxy

1. A DNS client considers the DNS proxy as the DNS server, and sends a DNS request to the DNS proxy. The destination address of the request is the IP address of the DNS proxy.
2. The DNS proxy searches the local static domain name resolution table and dynamic domain name resolution table after receiving the request. If the requested information is found, the DNS proxy returns a DNS reply to the client.
3. If the requested information is not found, the DNS proxy sends the request to the designated DNS server for domain name resolution.
4. After receiving a reply from the DNS server, the DNS proxy records the IP address-to-domain name mapping and forwards the reply to the DNS client.

With no DNS server or route to a DNS server specified, the DNS proxy does not forward DNS requests, or answer the requests from the DNS clients.

DNS spoofing

Figure 37 Application of DNS spoofing



DNS spoofing is applied to the dial-up network, as shown in [Figure 37](#).

- The device connects to the PSTN/ISDN network through a dial-up interface and triggers the establishment of a dial-up connection only when packets are to be forwarded through the dial-up interface.
- The device serves as a DNS proxy and is specified as a DNS server on the hosts. After the dial-up connection is established through the dial-up interface, the device dynamically obtains the DNS server address through DHCP or other autoconfiguration mechanisms.

Without DNS spoofing enabled, the device forwards the DNS requests received from the hosts to the DNS server, if it cannot find a match in the local domain name resolution table. However, without any dial-up connection established, the device cannot obtain the DNS server address and cannot forward or answer the requests from the clients. The domain name cannot be resolved and no traffic triggers the establishment of a dial-up connection.

DNS spoofing can solve the problem. DNS spoofing enables the device to reply the DNS client with a configured IP address when the device does not have a DNS server address or route to a DNS server. Subsequent packets sent by the DNS client trigger the establishment of a dial-up connection with the network.

In the network of [Figure 37](#), a host accesses the HTTP server in following these steps.

1. The host sends a DNS request to the device to resolve the domain name of the HTTP server into an IP address.
2. Upon receiving the request, the device searches the local static and dynamic DNS entries for a match. If no match is found and the device does know the DNS server address, the device spoofs the host by replying a configured IP address. The TTL of the DNS reply is 0. The device must have a route to the IP address with the dial-up interface as the outgoing interface.
3. Upon receiving the reply, the host sends an HTTP request to the replied IP address.
4. When forwarding the HTTP request through the dial-up interface, the device establishes a dial-up connection with the network and dynamically obtains the DNS server address through DHCP or other autoconfiguration mechanisms.
5. When the DNS reply ages out, the host sends a DNS request to the device again.
6. Then the device operates the same as a DNS proxy. For more information, see [“Operation of a DNS proxy.”](#)
7. After obtaining the IP address of the HTTP server, the host can access the HTTP server.

Because the IP address configured with DNS spoofing is not the actual IP address of the requested domain name, the TTL of the DNS reply is set to 0 to prevent the DNS client from generating incorrect domain name-to-IP address mappings.

Configuring the IPv4 DNS client

Configuring static domain name resolution

Configuring static domain name resolution refers to specifying the mappings between host names and IPv4 addresses. Static domain name resolution allows applications such as Telnet to contact hosts by using host names instead of IPv4 addresses.

To configure static domain name resolution:

| Step | Command | Remarks |
|---|------------------------------------|---|
| 1. Enter system view. | system-view | — |
| 2. Configure a mapping between a host name and an IPv4 address. | ip host hostname ip-address | Required. Not configured by default. |

The IPv4 address you last assign to the host name overwrites the previous one if there is any.

You can create up to 50 static mappings between domain names and IPv4 addresses.

Configuring dynamic domain name resolution

To send DNS queries to a correct server for resolution, dynamic domain name resolution needs to be enabled and a DNS server needs to be configured.

In addition, configure a DNS suffix that the system automatically adds to the provided domain name for resolution.

To configure dynamic domain name resolution:

| Step | Command | Remarks |
|---|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enable dynamic domain name resolution. | dns resolve | Required. Disabled by default. |
| 3. Specify a DNS server. | System view. dns server ip-address | Required. Not specified by default. |
| | Interface view. interface interface-type interface-number dns server ip-address | |
| | quit | |
| 4. Configure a DNS suffix. | dns domain domain-name | Optional. Not configured by default. Only the provided domain name is resolved. |

In system view, configure up to six DNS servers, including those with IPv6 addresses. The total number of DNS servers configured in interface view must be within six.

A DNS server configured in system view has a higher priority than one configured in interface view. A DNS server configured earlier has a higher priority than one configured later in the same view. A DNS server manually configured has a higher priority than one dynamically obtained through DHCP. A name query request is first sent to the DNS server that has the highest priority. If no reply is received, it is sent to the DNS server that has the second highest priority, and thus in turn.

Specify up to ten DNS suffixes.

Configuring the DNS proxy

| Step | Command | Remarks |
|--------------------------|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enable DNS proxy. | dns proxy enable | Required. Disabled by default. |
| 3. Specify a DNS server. | System view. dns server ip-address | Required. |
| | Interface view. interface interface-type interface-number | Configure the DNS server in at least one view. |
| | dns server ip-address | No DNS server is specified by default. |

Specify multiple DNS servers by using **dns server** repeatedly. Upon receiving a name query request from a client, the DNS proxy forwards the request to the specified DNS servers in turn. The DNS server configured in system view has a higher priority than the DNS server configured in interface view. That is, the request is first forwarded to each DNS server configured in system view, and, if no reply is obtained, then sent to each DNS server configured in interface view in turn.

Configuring DNS spoofing

Configuration prerequisites

DNS spoofing is effective only when:

- The DNS proxy is enabled on the switch.
- No DNS server or route to any DNS server is specified on the switch.

Configuration procedure

| Step | Command | Remarks |
|---|--------------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable DNS spoofing and specify the translated IP address. | dns spoofing ip-address | Required. Disabled by default. |

Displaying and maintaining IPv4 DNS

| Task | Command | Remarks |
|--|--|------------------------|
| Display the static IPv4 domain name resolution table. | <code>display ip host [{ begin exclude include } regular-expression]</code> | Available in any view |
| Display IPv4 DNS server information. | <code>display dns server [dynamic] [{ begin exclude include } regular-expression]</code> | Available in any view |
| Display DNS suffixes. | <code>display dns domain [dynamic] [{ begin exclude include } regular-expression]</code> | Available in any view |
| Display the information of the dynamic IPv4 domain name cache. | <code>display dns host ip [{ begin exclude include } regular-expression]</code> | Available in any view |
| Clear the information of the dynamic IPv4 domain name cache. | <code>reset dns host ip</code> | Available in user view |

IPv4 DNS configuration examples

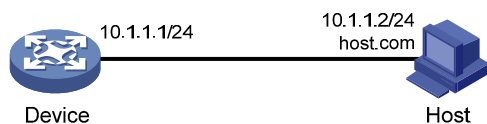
Static domain name resolution configuration example

Network requirements

As shown in [Figure 38](#), the device wants to access the host by using an easy-to-remember domain name rather than an IP address.

Configure static domain name resolution on the device so that the device can use the domain name `host.com` to access the host whose IP address is `10.1.1.2`.

Figure 38 Network diagram for static domain name resolution



Configuration procedure

Configure a mapping between host name `host.com` and IP address `10.1.1.2`.

```
<Sysname> system-view
[Sysname] ip host host.com 10.1.1.2
```

Use **ping host.com** to verify that the device can use static domain name resolution to resolve domain name `host.com` into IP address `10.1.1.2`.

```
[Sysname] ping host.com
PING host.com (10.1.1.2):
 56 data bytes, press CTRL_C to break
 Reply from 10.1.1.2: bytes=56 Sequence=1 ttl=128 time=1 ms
 Reply from 10.1.1.2: bytes=56 Sequence=2 ttl=128 time=4 ms
 Reply from 10.1.1.2: bytes=56 Sequence=3 ttl=128 time=3 ms
 Reply from 10.1.1.2: bytes=56 Sequence=4 ttl=128 time=2 ms
 Reply from 10.1.1.2: bytes=56 Sequence=5 ttl=128 time=3 ms
```

```
--- host.com ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/2/4 ms
```

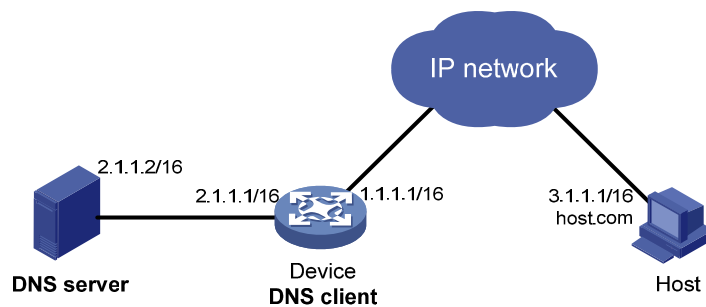
Dynamic domain name resolution configuration example

Network requirements

As shown in [Figure 39](#), the device wants to access the host by using an easy-to-remember domain name rather than an IP address, and to request the DNS server on the network for an IP address by using dynamic domain name resolution. The IP address of the DNS server is 2.1.1.2/16 and the DNS server has a com domain, which stores the mapping between domain name host and IP address 3.1.1.1/16.

Configure dynamic domain name resolution and the domain name suffix com on the device that serves as a DNS client so that the device can use domain name host to access the host with the domain name host.com and the IP address 3.1.1.1/16.

Figure 39 Network diagram for dynamic domain name resolution



Configuration procedure

Before performing the following configuration, make sure that the device and the host are accessible to each other via available routes, and the IP addresses of the interfaces are configured as shown [Figure 39](#).

This configuration can vary with different DNS servers. The following configuration is performed on a Windows server 2000 PC.

1. Configure the DNS server

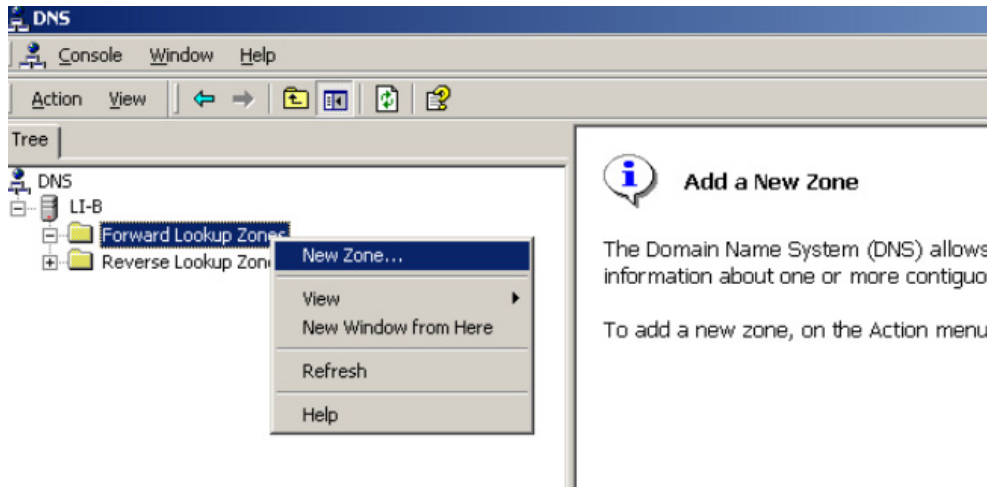
Enter the DNS server configuration page.

Select **Start > Programs > Administrative Tools > DNS**.

Create zone com.

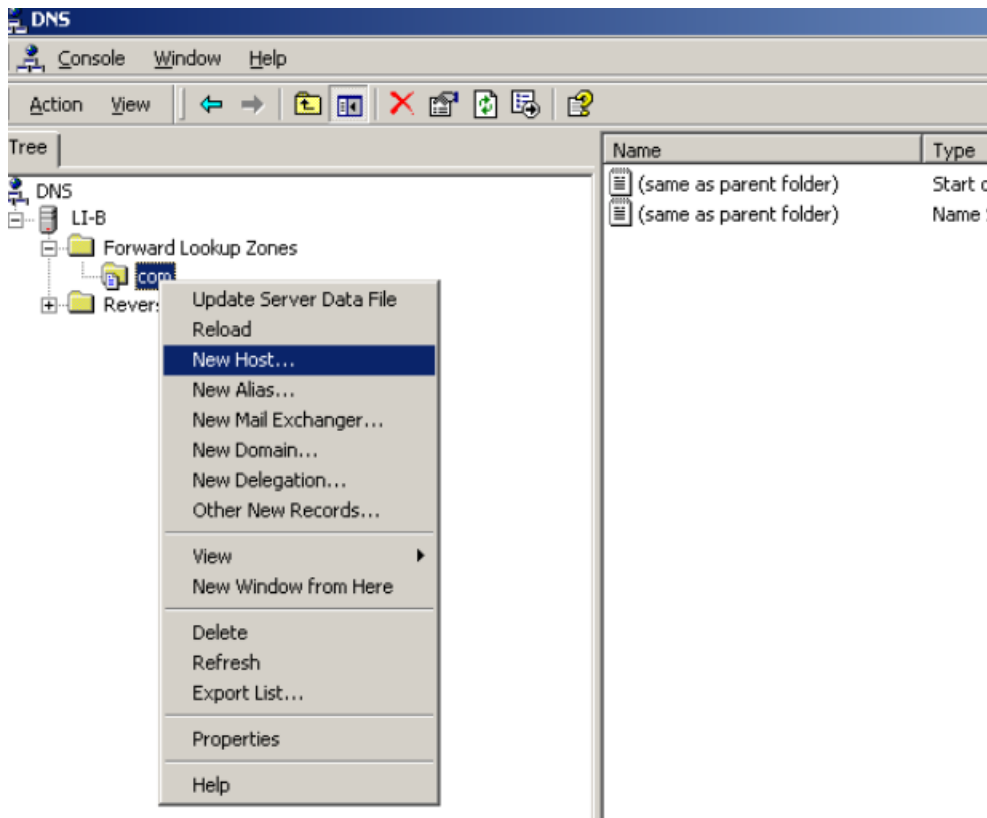
As shown in [Figure 40](#), right click **Forward Lookup Zones**, select **New zone**, and then follow the instructions to create a new zone named **com**.

Figure 40 Create a zone



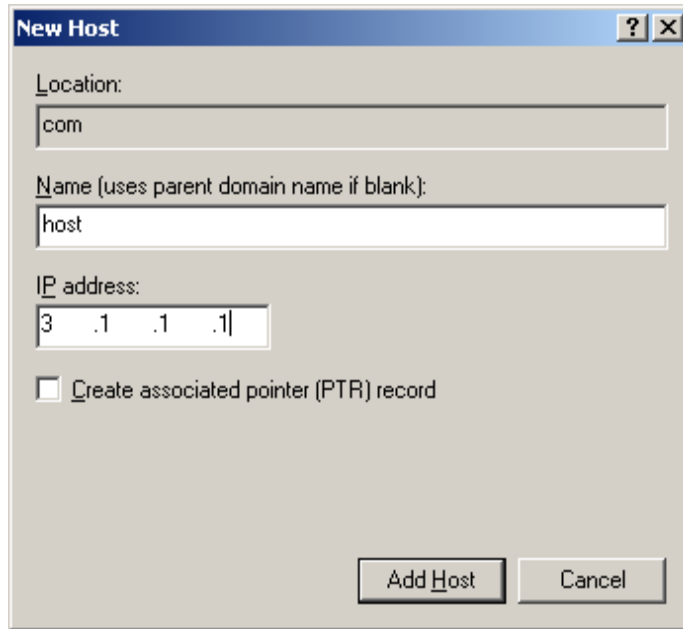
Create a mapping between host name and IP address.

Figure 41 Add a host



In [Figure 41](#), right click zone **com**, and then select **New Host** to bring up a dialog box as shown in [Figure 42](#). Enter host name **host** and IP address **3.1.1.1**.

Figure 42 Add a mapping between domain name and IP address



2. Configure the DNS client

Enable dynamic domain name resolution.

```
<Sysname> system-view  
[Sysname] dns resolve
```

Specify the DNS server 2.1.1.2.

```
[Sysname] dns server 2.1.1.2
```

Configure com as the name suffix.

```
[Sysname] dns domain com
```

3. Configuration verification

Use **ping host** on the device to verify that the communication between the device and the host is normal and that the corresponding destination IP address is 3.1.1.1.

```
[Sysname] ping host  
Trying DNS resolve, press CTRL_C to break  
Trying DNS server (2.1.1.2)  
PING host.com (3.1.1.1):  
56 data bytes, press CTRL_C to break  
Reply from 3.1.1.1: bytes=56 Sequence=1 ttl=126 time=3 ms  
Reply from 3.1.1.1: bytes=56 Sequence=2 ttl=126 time=1 ms  
Reply from 3.1.1.1: bytes=56 Sequence=3 ttl=126 time=1 ms  
Reply from 3.1.1.1: bytes=56 Sequence=4 ttl=126 time=1 ms  
Reply from 3.1.1.1: bytes=56 Sequence=5 ttl=126 time=1 ms
```

```
--- host.com ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/1/3 ms
```

DNS proxy configuration example

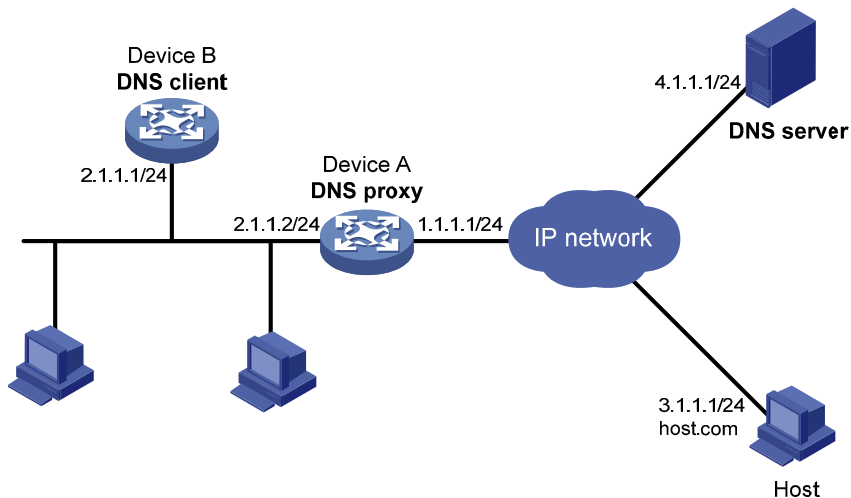
Network requirements

When the IP address of the DNS server changes, you must configure the new IP address of the DNS server on each device on the LAN. To simplify network management, use the DNS proxy function.

As shown in [Figure 43](#), specify Device A as the DNS server of Device B (the DNS client). Device A acts as a DNS proxy. The IP address of the real DNS server is 4.1.1.1.

Configure the IP address of the DNS proxy on Device B. DNS requests of Device B are forwarded to the real DNS server through the DNS proxy.

Figure 43 Network diagram for DNS proxy



Configuration procedure

Before performing the following configuration, assume that Device A, the DNS server, and the host are reachable to each other and the IP addresses of the interfaces are configured as shown in [Figure 43](#).

1. Configure the DNS server

This configuration can vary with different DNS servers. When a Windows server 2000 PC acts as the DNS server, see [“Dynamic domain name resolution configuration example”](#) for related configuration information.

2. Configure the DNS proxy

Specify the DNS server 4.1.1.1.

```
<DeviceA> system-view
[DeviceA] dns server 4.1.1.1
```

```
# Enable DNS proxy.
```

```
[DeviceA] dns proxy enable
```

3. Configure the DNS client

```
# Enable the domain name resolution function.
```

```
<DeviceB> system-view
```

```
[DeviceB] dns resolve
```

```
# Specify the DNS server 2.1.1.2.
```

```
[DeviceB] dns server 2.1.1.2
```

4. Configuration verification

```
# Execute ping host.com on Device B to verify that the communication between the device and the host is normal and that the corresponding destination IP address is 3.1.1.1.
```

```
[DeviceB] ping host.com
```

```
Trying DNS resolve, press CTRL_C to break
```

```
Trying DNS server (2.1.1.2)
```

```
PING host.com (3.1.1.1):
```

```
56 data bytes, press CTRL_C to break
```

```
Reply from 3.1.1.1: bytes=56 Sequence=1 ttl=126 time=3 ms
```

```
Reply from 3.1.1.1: bytes=56 Sequence=2 ttl=126 time=1 ms
```

```
Reply from 3.1.1.1: bytes=56 Sequence=3 ttl=126 time=1 ms
```

```
Reply from 3.1.1.1: bytes=56 Sequence=4 ttl=126 time=1 ms
```

```
Reply from 3.1.1.1: bytes=56 Sequence=5 ttl=126 time=1 ms
```

```
--- host.com ping statistics ---
```

```
5 packet(s) transmitted
```

```
5 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 1/1/3 ms
```

Troubleshooting IPv4 DNS configuration

Symptom

After enabling the dynamic domain name resolution, the user cannot get the correct IP address.

Solution

- Use **display dns host ip** to verify that the specified domain name is in the cache.
- If the specified domain name does not exist, check that dynamic domain name resolution is enabled and the DNS client can communicate with the DNS server.
- If the specified domain name is in the cache, but the IP address is incorrect, check that the DNS client has the correct IP address of the DNS server.
- Verify the mapping between the domain name and IP address is correct on the DNS server.

IPv6 DNS configuration

IPv6 DNS is responsible for translating domain names into IPv6 addresses. Similar to IPv4 DNS, IPv6 DNS involves static domain name resolution and dynamic domain name resolution. The functions and implementations of the two types of domain name resolution are the same as those of IPv4 DNS. For more information, see the chapter “IPv4 DNS configuration.”

Configuring the IPv6 DNS client

Configuring static domain name resolution

Configuring static domain name resolution refers to specifying the mappings between host names and IPv6 addresses. Static domain name resolution allows applications such as Telnet to contact hosts by using host names instead of IPv6 addresses.

To configure static domain name resolution:

| Step | Command | Remarks |
|---|--|---|
| 1. Enter system view. | system-view | — |
| 2. Configure a mapping between a host name and an IPv6 address. | ipv6 host hostname ipv6-address | Required. Not configured by default. |

A host name can only be mapped to one IPv6 address. If you map a host name to different IPv6 addresses, the last configuration takes effect.

Configure up to 50 mappings between domain name and IPv6 address.

Configuring dynamic domain name resolution

To send DNS queries to a correct server for resolution, dynamic domain name resolution must be enabled and a DNS server must be configured.

In addition, configure a DNS suffix that the system automatically adds to the provided domain name for resolution.

To configure dynamic domain name resolution:

| Step | Command | Remarks |
|---|--|---|
| 1. Enter system view. | system-view | — |
| 2. Enable dynamic domain name resolution. | dns resolve | Required. Disabled by default. |
| 3. Specify a DNS server. | dns server ipv6 ipv6-address [interface-type interface-number] | Required. Not specified by default. If the IPv6 address of a DNS server is a link-local address, you need to specify the interface-type and interface-number arguments. |
| 4. Configure a DNS suffix. | dns domain domain-name | Required. Not configured by default. Only the provided domain name is resolved. |

Dns resolve and **dns domain** are the same as those of IPv4 DNS.

Configure up to six DNS servers, including those with IPv4 addresses.

Specify up to ten DNS suffixes.

Displaying and maintaining IPv6 DNS

| Task | Command | Remarks |
|--|--|------------------------|
| Display the static IPv6 domain name resolution table. | display ipv6 host [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display IPv6 DNS server information. | display dns ipv6 server [dynamic] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display DNS suffixes. | display dns domain [dynamic] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display the information of dynamic IPv6 domain name cache. | display dns host ipv6 [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Clear the information of dynamic IPv6 domain name cache. | reset dns host ipv6 | Available in user view |

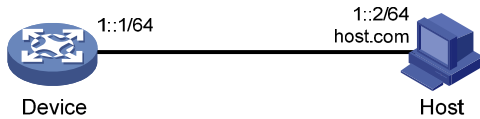
IPv6 DNS configuration examples

Static domain name resolution configuration example

Network requirements

As shown in [Figure 44](#), the device wants to access the host by using an easy-to-remember domain name rather than an IPv6 address. Configure static domain name resolution on the device so that the device can use the domain name host.com to access the host whose IPv6 address is 1::2.

Figure 44 Network diagram for static domain name resolution



Configuration procedure

Configure a mapping between host name host.com and IPv6 address 1::2.

```
<Device> system-view
[Device] ipv6 host host.com 1::2
```

Enable IPv6 packet forwarding.

```
[Device] ipv6
```

Use **ping ipv6 host.com** to verify that the device can use static domain name resolution to resolve domain name host.com into IPv6 address 1::2.

```
[Device] ping ipv6 host.com
PING host.com (1::2):
 56 data bytes, press CTRL_C to break
  Reply from 1::2:
    bytes=56 Sequence=1 hop limit=128  time = 3 ms
  Reply from 1::2:
    bytes=56 Sequence=2 hop limit=128  time = 1 ms
  Reply from 1::2:
    bytes=56 Sequence=3 hop limit=128  time = 1 ms
  Reply from 1::2:
    bytes=56 Sequence=4 hop limit=128  time = 2 ms
  Reply from 1::2:
    bytes=56 Sequence=5 hop limit=128  time = 2 ms
--- host.com ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/1/3 ms
```

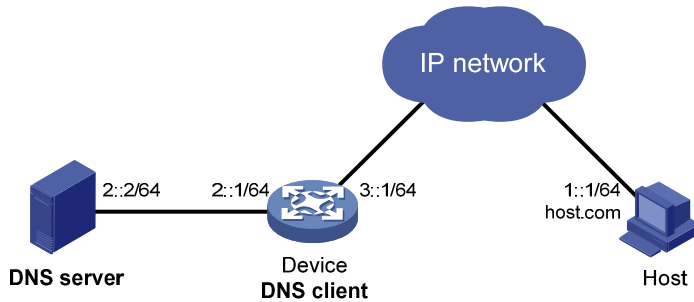
Dynamic domain name resolution configuration example

Network requirements

As shown in [Figure 45](#), the device wants to access the host by using an easy-to-remember domain name rather than an IPv6 address. The IPv6 address of the DNS server is 2::2/64 and the server has a com domain, which stores the mapping between domain name host and IPv6 address 1::1/64.

Configure dynamic domain name resolution and the domain name suffix com on the device that serves as a DNS client so that the device can use domain name host to access the host with the domain name host.com and the IPv6 address 1::1/64.

Figure 45 Network diagram of dynamic domain name resolution



Configuration procedure

Before performing the following configuration, make sure that the device and the host are accessible to each other via available routes, and the IPv6 addresses of the interfaces are configured as shown Figure 45.

This configuration can vary with different DNS servers. The following configuration is performed on a PC running Windows server 2003. Make sure that the DNS server supports the IPv6 DNS function so that the server can process IPv6 DNS packets, and the interfaces of the DNS server can forward IPv6 packets.

1. Configure the DNS server

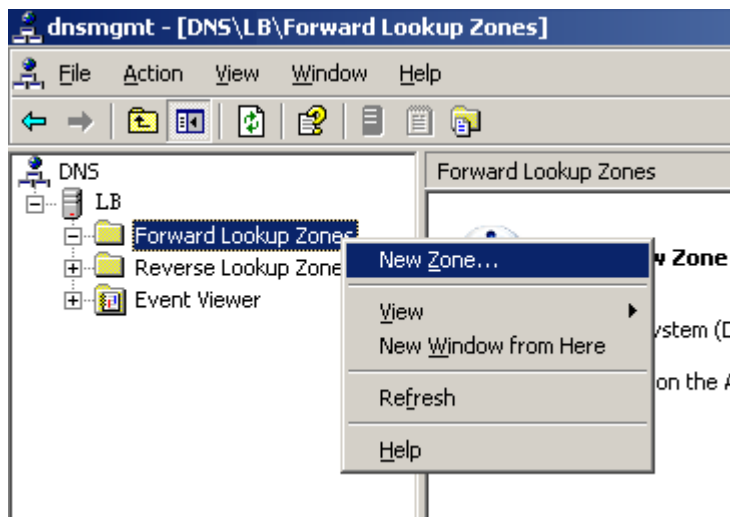
Enter the DNS server configuration page.

Select **Start > Programs > Administrative Tools > DNS**.

Create zone com.

As shown in Figure 46, right click **Forward Lookup Zones**, select **New zone**, and then follow the instructions to create a new zone named **com**.

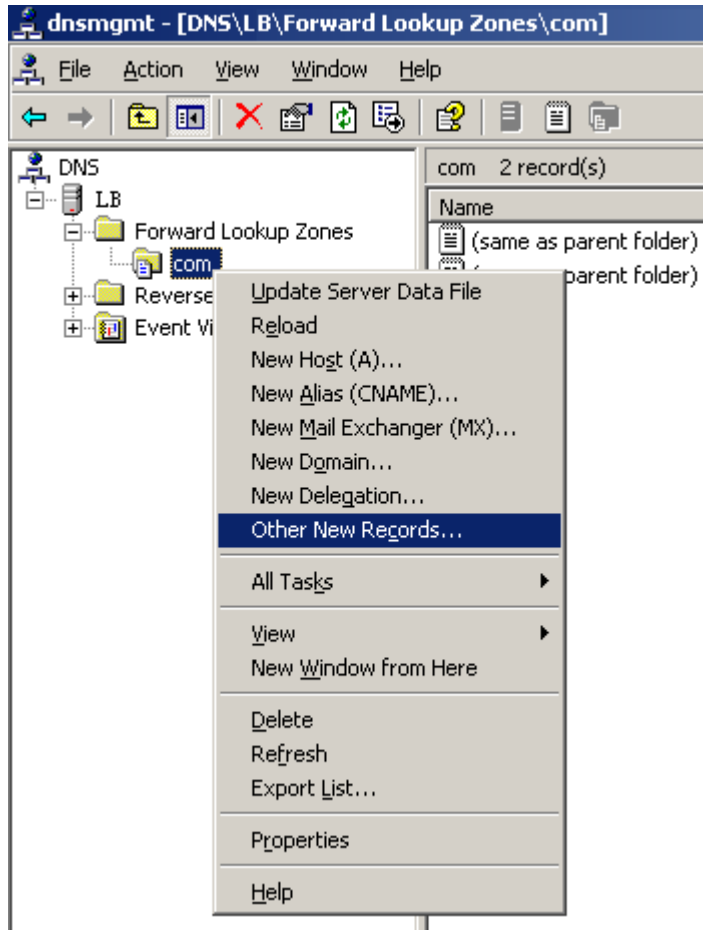
Figure 46 Create a zone



Create a mapping between the host name and the IPv6 address.

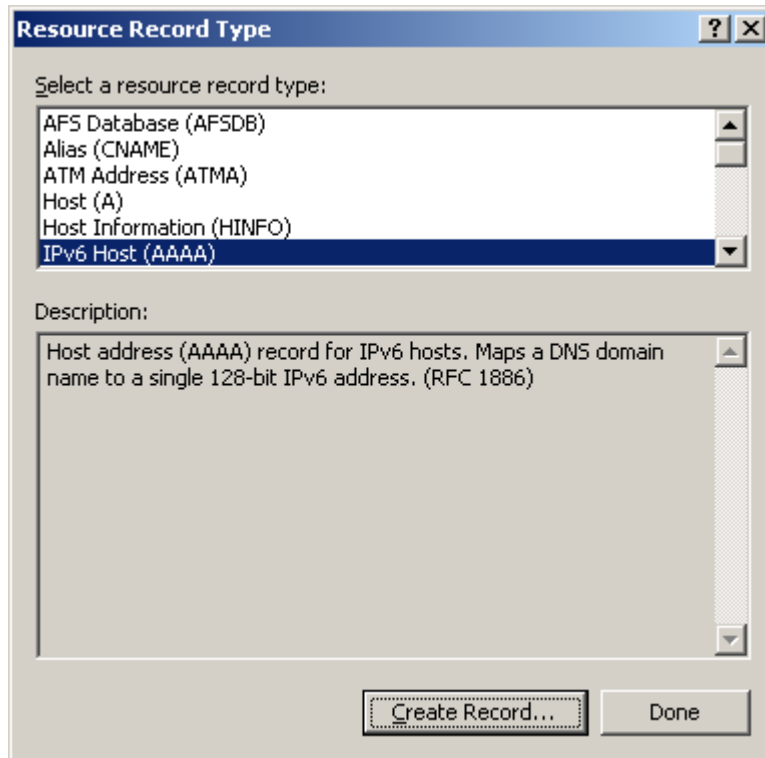
As shown in Figure 47, right click zone **com**.

Figure 47 Create a record



In Figure 47, select **Other New Records** to bring up a dialog box as shown in Figure 48. Select **IPv6 Host (AAA)** as the resource record type.

Figure 48 Select the resource record type



As shown in [Figure 49](#), type host name host and IPv6 address 1::1, and then click **OK**.

Figure 49 Add a mapping between domain name and IPv6 address

The screenshot shows a 'New Resource Record' dialog box with the following fields and values:

- Host (uses parent domain if left blank): host
- Fully qualified domain name (FQDN): host.com.
- IP version 6 host address: 1::1

Buttons: OK, Cancel

2. Configure the DNS client

Enable dynamic domain name resolution.

```
<Device> system-view  
[Device] dns resolve
```

Specify the DNS server 2::2.

```
[Device] dns server ipv6 2::2
```

Configure com as the DNS suffix.

```
[Device] dns domain com
```

3. Configuration verification

Use **ping ipv6 host** on the device to verify that the communication between the device and the host is normal and that the corresponding destination IP address is 1::1.

```
[Device] ping ipv6 host
Trying DNS resolve, press CTRL_C to break
Trying DNS server (2::2)
PING host.com (1::1):
56 data bytes, press CTRL_C to break
  Reply from 1::1
    bytes=56 Sequence=1 hop limit=126  time = 2 ms
  Reply from 1::1
    bytes=56 Sequence=2 hop limit=126  time = 1 ms
  Reply from 1::1
    bytes=56 Sequence=3 hop limit=126  time = 1 ms
  Reply from 1::1
    bytes=56 Sequence=4 hop limit=126  time = 1 ms
  Reply from 1::1
    bytes=56 Sequence=5 hop limit=126  time = 1 ms

--- host.com ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
round-trip min/avg/max = 1/1/2 ms
```

IP performance optimization configuration

Use the following configurations to optimize IP performance:

- Enabling the switch to receive and forward directed broadcasts
- Configuring the TCP send/receive buffer size
- Configuring TCP timers
- Enabling ICMP error packets sending
- Enabling support for ICMP extensions

Enabling reception and forwarding of directed broadcasts to a directly connected network

Directed broadcast packets are broadcast on a specific network. In the destination IP address of a directed broadcast, the network ID identifies the target network, and the host ID is made up of all ones. If the switch is allowed to forward directed broadcasts to a directly connected network, hackers can mount attacks to the network. However, enable the feature when using the following functions:

- Using the UDP Helper function to convert broadcasts to unicasts and forward them to a specified server.
- Using the Wake on LAN function to forward directed broadcasts to a host on the remote network.

Enabling reception of directed broadcasts to a directly connected network

If the switch is enabled to receive directed broadcasts, the switch determines whether to forward them according to the configuration on the outgoing interface.

To enable the switch to receive directed broadcasts:

| Step | Command | Remarks |
|--|-----------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable the switch to receive directed broadcasts. | ip forward-broadcast | Required. Disabled by default. |

Enabling forwarding of directed broadcasts to a directly connected network

| Step | Command | Remarks |
|---|--|-----------------------------------|
| 1. Enter system view. | <code>system-view</code> | — |
| 2. Enter interface view. | <code>interface interface-type interface-number</code> | — |
| 3. Enable the interface to forward directed broadcasts. | <code>ip forward-broadcast [acl acl-number]</code> | Required. Disabled by default. |

If an ACL is referenced in `ip forward-broadcast`, only packets permitted by the ACL can be forwarded.

If you repeatedly execute `ip forward-broadcast` on an interface, only the last executed command takes effect. If the command executed last does not include the `acl acl-number`, the ACL configured previously is removed.

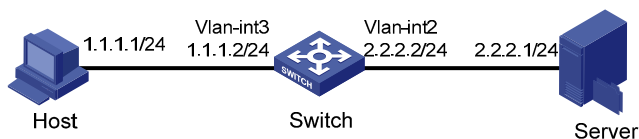
Configuration example

Network requirements

As shown in Figure 50, the host's interface and VLAN-interface 3 of the switch are on the same network segment (1.1.1.0/24). VLAN-interface 2 of Switch and the server are on another network segment (2.2.2.0/24). The default gateway of the host is VLAN-interface 3 (IP address 1.1.1.2/24) of Switch.

Configure the switch so that the server can receive directed broadcasts from the host to IP address 2.2.2.255.

Figure 50 Network diagram for receiving and forwarding directed broadcasts



Configuration procedure

- Configure Switch A

Enable Switch A to receive directed broadcasts.

```
<SwitchA> system-view
[SwitchA] ip forward-broadcast
```

Configure IP addresses for VLAN-interface 3 and VLAN-interface 2.

```
[SwitchA] interface vlan-interface 3
[SwitchA-Vlan-interface3] ip address 1.1.1.2 24
[SwitchA-Vlan-interface3] quit
[SwitchA] interface vlan-interface 2
[SwitchA-Vlan-interface2] ip address 2.2.2.2 24
```

```
# Enable VLAN-interface 2 to forward directed broadcasts.  
[Switch-Vlan-interface2] ip forward-broadcast
```

Configuring TCP attributes

Configuring the TCP send/receive buffer size

| Step | Command | Remarks |
|---|-------------------------------------|-------------------------------|
| 1. Enter system view. | <code>system-view</code> | — |
| 2. Configure the size of TCP receive/send buffer. | <code>tcp window window-size</code> | Optional. 8 KB by default. |

Configuring TCP timers

CAUTION:

The actual length of the finwait timer is determined by the following formula:

Actual length of the finwait timer = (Configured length of the finwait timer – 75) + configured length of the synwait timer

Configure the following TCP timers:

- **Synwait timer**—When sending a SYN packet, TCP starts the synwait timer. If no response packet is received within the synwait timer interval, the TCP connection cannot be created.
- **Finwait timer**—When a TCP connection is changed into FIN_WAIT_2 state, the finwait timer is started. If no FIN packet is received within the timer interval, the TCP connection is terminated. If a FIN packet is received, the TCP connection state changes to TIME_WAIT. If a non-FIN packet is received, the system restarts the timer upon receiving the last non-FIN packet. The connection is broken after the timer expires.

To configure TCP timers:

| Step | Command | Remarks |
|-------------------------------------|---|--------------------------------------|
| 1. Enter system view. | <code>system-view</code> | — |
| 2. Configure the TCP synwait timer. | <code>tcp timer syn-timeout time-value</code> | Optional. 75 seconds by default. |
| 3. Configure the TCP finwait timer. | <code>tcp timer fin-timeout time-value</code> | Optional. 675 seconds by default. |

Configuring ICMP to send error packets

Sending error packets is a major function of ICMP. In case of network abnormalities, error packets are usually sent by the network or transport layer protocols to notify corresponding devices so as to facilitate control and management.

Advantages of sending ICMP error packets

ICMP error packets include redirect, timeout, and destination unreachable packets.

1. Sending ICMP redirect packets

A host can have only a default route to the default gateway in its routing table after startup. The default gateway sends ICMP redirect packets to the source host, telling it to reselect a correct next hop to send the subsequent packets, if the following conditions are satisfied:

- The receiving and forwarding interfaces are the same.
- The selected route has not been created or modified by an ICMP redirect packet.
- The selected route is not the default route of the switch.
- There is no source route option in the packet.

The ICMP redirect packets function simplifies host administration and enables a host to gradually establish a sound routing table to find the best route.

2. Sending ICMP timeout packets

If the switch received an IP packet with a timeout error, it drops the packet and sends an ICMP timeout packet to the source.

The switch sends an ICMP timeout packet under the following conditions:

- If the switch finds the destination of a packet is not itself and the TTL field of the packet is 1, it sends a "TTL timeout" ICMP error message.
- When the switch receives the first fragment of an IP datagram whose destination is the switch itself, it starts a timer. If the timer times out before all fragments of the datagram are received, the switch sends a "reassembly timeout" ICMP error packet.

3. Sending ICMP destination unreachable packets

If the switch receives an IP packet with the destination unreachable, it drops the packet and sends an ICMP destination unreachable error packet to the source.

Conditions for sending this ICMP packet:

- If neither a route nor the default route for forwarding a packet is available, the switch sends a "network unreachable" ICMP error packet.
- If the destination of a packet is local but the transport layer protocol of the packet is not supported by the local device, the switch sends a "protocol unreachable" ICMP error packet to the source.
- When receiving a packet with the destination being local and transport layer protocol being UDP, if the packet's port number does not match the running process, the switch sends the source a "port unreachable" ICMP error packet.
- If the source uses "strict source routing" to send packets, but the intermediate device finds that the next hop specified by the source is not directly connected, the switch sends the source a "source routing failure" ICMP error packet.

- When forwarding a packet, if the MTU of the sending interface is smaller than the packet but the packet has been set as “Don’t Fragment”, the switch sends the source a “fragmentation needed and DF-set” ICMP error packet.

Disadvantages of sending ICMP error packets

Sending ICMP error packets facilitates network control and management, but it still has the following disadvantages:

- Sending a lot of ICMP packets increases network traffic.
- If a switch receives a lot of malicious packets that cause it to send ICMP error packets, its performance are reduced.
- As the redirection function increases the routing table size of a host, the host’s performance are reduced if its routing table becomes very large.
- If an attacker sends abnormal traffic that causes the switch to generate ICMP destination unreachable packets, end users can be affected.

To prevent such problems, disable the switch from sending ICMP error packets.

Configuration procedure

| Step | Command | Remarks |
|--|------------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable sending of ICMP redirect packets. | ip redirects enable | Required. Disabled by default. |
| 3. Enable sending of ICMP timeout packets. | ip ttl-expires enable | Required. Disabled by default. |
| 4. Enable sending of ICMP destination unreachable packets. | ip unreachable enable | Required. Disabled by default. |

The switch stops sending “TTL timeout” ICMP error packets after sending ICMP timeout packets is disabled. However, “reassembly timeout” error packets are sent normally.

Enabling ICMP extension support

ICMP messages are of a fixed format and cannot carry extension information. With support for ICMP extensions enabled, a switch appends an extension information field to the ICMP messages as needed. The switch can append only MPLS label information to ICMP messages.

ICMP extensions for MPLS

In MPLS networks, when a packet's TTL expires, MPLS strips the MPLS header, encapsulates the remaining datagram into an ICMP time exceeded message, and sends the message to the egress router of the MPLS tunnel. Then the egress router sends the message back to the ingress router of the tunnel. The ICMP message, however, does not contain the label information that is very important to the ingress router. With support for ICMP extensions enabled, the switch appends the MPLS label to the ICMP time exceeded message before sending it back to the ingress router of the tunnel.

ICMP extensions are usually used for an enhanced traceroute implementation in MPLS networks, in which MPLS label information of each hop the original datagram arrives at is printed.

Handling ICMP messages

ICMP messages can be classified into the following types:

- **Common ICMP messages**—Without any extension information.
- **Extended ICMP messages with a length field**—Carry extension information and a length field. The length field indicates the length of the original datagram that is encapsulated within the ICMP header and excludes the ICMP extension length. This form of ICMP message complies with RFC 4884.
- **Extended ICMP messages without a length field**—Carry extension information but does not contain a length field. This form of ICMP message does not comply with RFC 4884.

Based on how these messages are handled, the switch can work in one of these modes: common mode, compliant mode, and non-compliant mode. Table 6 shows how ICMP messages are handled in different working modes.

Table 6 Handling ICMP messages

| Device mode | ICMP messages sent | ICMP messages received | Remarks |
|--------------------|---|--|--|
| Common mode | Common ICMP messages | Common ICMP messages | Extension information in extended ICMP messages are not processed. |
| Compliant mode | Common ICMP messages Extended ICMP messages with a length field | Common ICMP messages Extended ICMP messages with a length field | Extended ICMP messages without a length field are handled as common ICMP messages. |
| Non-compliant mode | Common ICMP messages Extended ICMP messages without a length field | All three types of ICMP messages | — |

ICMP/ICMPv6 messages that can carry extension information include only IPv4 redirect messages, IPv4/IPv6 time exceeded messages, and IPv4/IPv6 destination unreachable messages.

Configuration procedure

Enabling support for ICMP extensions

| Step | Command | Remarks |
|--|---|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable support for ICMP extensions in compliant mode. | ip icmp-extensions compliant | Optional. Disabled by default. |
| 3. Enable support for ICMP extensions in non-compliant mode. | ip icmp-extensions non-compliant | Optional. Disabled by default. |

When support for ICMP extensions is disabled, no ICMP message sent by the switch contains extension information.

Displaying and maintaining IP performance optimization

| Task | Command | Remarks |
|--|--|------------------------|
| Display TCP connection statistics. | display tcp statistics [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display UDP statistics. | display udp statistics [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display statistics of IP packets. | display ip statistics [slot <i>slot-number</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display ICMP statistics. | display icmp statistics [slot <i>slot-number</i>] [{ begin exclude include } <i>regular-expression</i>] <i>regular-expression</i>] | Available in any view |
| Display socket information. | display ip socket [socktype <i>sock-type</i>] [<i>task-id</i> <i>socket-id</i>] [slot <i>slot-number</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display FIB information. | display fib [vpn-instance <i>vpn-instance-name</i>] [acl <i>acl-number</i> ip-prefix <i>ip-prefix-name</i>] [{ begin include exclude } <i>regular-expression</i>] | Available in any view |
| Display FIB information matching the specified destination IP address. | display fib [vpn-instance <i>vpn-instance-name</i>] <i>ip-address</i> [<i>mask</i> <i>mask-length</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Clear statistics of IP packets. | reset ip statistics [slot <i>slot-number</i>] | Available in user view |
| Clear statistics of TCP connections. | reset tcp statistics | Available in user view |
| Clear statistics of UDP traffic. | reset udp statistics | Available in user view |

IRDP configuration

As an extension of the ICMP, the IRDP enables hosts to discover the IP addresses of their neighboring routers and set their default routes.

The hosts in this document support IRDP.

Before a host can send packets to another network, it must know the IP address of at least one router on the local subnet. The host can obtain this information either through manual configuration, or from routing protocol packets sent by routers on the local subnet.

Both methods have disadvantages. The first method requires the administrator to manually configure and maintain router address information on hosts, and cannot track dynamic changes. The second method requires hosts to recognize various routing protocols, and fails to work if no routing protocol runs on the local subnet.

IRDP was introduced to solve the problem. IRDP uses two new types of ICMP messages to allow hosts to discover neighboring routers. IRDP adapts to dynamic changes, requires less manual configuration, and does not rely on any routing protocols.

Working mechanism

IRDP uses the following types of ICMP messages.

- **RA**—Sent by a router to advertise its IP address and preference.
- **RS**—Sent by a host to voluntarily request the IP addresses of routers on the subnet.

IRDP works with the following steps.

- A router periodically broadcasts or multicasts an RA, which contains the IP addresses (including the primary IP address and manually configured secondary IP addresses) of interfaces. Hosts listen for RAs to obtain the IP addresses of neighboring routers.
- Rather than wait for RAs, a newly attached host can voluntarily send an RS to request immediate RAs for the IP addresses of routers on the subnet. If no response to the RS is received, the host retransmits the RS several times. If the host still receives no RAs, it obtains the IP addresses of routers from periodic RAs.
- Upon receiving an RA, a host adds the IP addresses in the RA to its routing table. The host selects the IP address with the highest preference among all obtained IP addresses as the default gateway.

IRDP allows hosts to locate routers, but does not suggest the best route to a specific destination. If a host selects a router that is not the best next hop to a specific destination, the router sends back an ICMP redirect message to provide a better next hop.

Terminology

IP address preference

Every IP address advertised in RAs has a preference value. The IP address with the highest preference is selected as the default router address.

Configure the preference for IP addresses advertised on a router interface.

The bigger the preference value, the higher the preference. The minimum preference value (-2147483648) is used to indicate that the address, even though it can be advertised, is not to be used by neighboring hosts as a default router address.

IP address lifetime

An RA contains a lifetime field that specifies the lifetime of advertised IP addresses. If no new RA for an IP address is received within the lifetime of the IP address, the host removes the corresponding route information.

All the IP addresses advertised by an interface have the same lifetime.

Advertising interval

A router interface with IRDP enabled sends out RAs at a random interval between the minimum advertising interval and the maximum advertising interval. This mechanism prevents the local link from being overloaded due to a large number of RAs sent simultaneously from routers.

HP recommends you shorten the advertising interval on a link that suffers high packet loss rates.

RA destination address

An RA uses either of the following destination IP addresses:

- Broadcast address 255.255.255.255
- Multicast address 224.0.0.1, which identifies all hosts on the local subnet.

By default, the destination IP address of an RA is the broadcast address. If the interface that sends RAs supports multicast, configure 224.0.0.1 as the destination IP address.

Proxy-advertised IP addresses

By default, an interface advertises its primary IP address and manually configured secondary IP addresses. configure other IP addresses for an interface to proxy-advertise.

Protocols and standards

- RFC 1256, *ICMP Router Discovery Messages*

Configuring IRDP

| Step | Command | Remarks |
|--|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface <i>interface-type</i> <i>interface-number</i> | The interface can be a Layer 3 Ethernet interface or VLAN interface. |
| 3. Enable IRDP on the interface. | ip irdp | Required. Disabled by default. |
| 4. Configure the preference of advertised IP addresses. | ip irdp preference <i>preference-value</i> | Optional. The preference defaults to 0. The specified preference applies to all advertised IP addresses, including the primary IP address and the manually configured secondary IP addresses of the interface. |
| 5. Set the lifetime of advertised IP addresses. | ip irdp lifetime <i>life-number</i> | Optional. 1800 seconds by default. The specified lifetime applies to all advertised IP addresses, including the IP address of the interface and proxy-advertised IP addresses on the interface. |
| 6. Set the minimum advertising interval. | ip irdp minadvertinterval <i>min-value</i> | Optional. 450 seconds by default. |
| 7. Set the maximum advertising interval. | ip irdp maxadvertinterval <i>max-value</i> | Optional. 600 seconds by default. |
| 8. Configure the multicast address (224.0.0.1) as the destination IP address of RAs. | ip irdp multicast | Optional. By default, RAs use the broadcast address 255.255.255.255 as the destination IP address. |
| 9. Specify a proxy-advertised IP address and its preference. | ip irdp address <i>ip-address</i> <i>preference</i> | Optional. |

IRDP configuration only takes effect when IRDP is enabled.

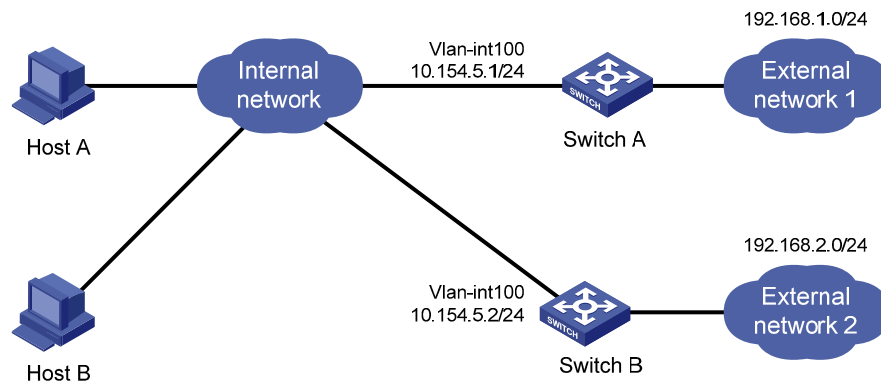
IRDP configuration example

Network requirements

Host A and Host B that run Linux systems reside in the internal network of a company. Switch A and Switch B serve as the egress routers and connect to external networks 192.168.1.0/24 and 192.168.2.0/24 respectively.

It is required to configure Switch A as the default router of the hosts. The packets to the external networks can be properly routed.

Figure 51 Network diagram for IRDP configuration



Configuration procedure

1. Configure Switch A

Specify the IP address for Vlan-interface100.

```
<SwitchA> system-view
```

```
[SwitchA] interface Vlan-interface 100
```

```
[SwitchA-Vlan-interface100] ip address 10.154.5.1 24
```

Enable IRDP on Vlan-interface 100.

```
[SwitchA-Vlan-interface100] ip irdp
```

Specify preference 1000 for the IP address of Vlan-interface 100.

```
[SwitchA-Vlan-interface100] ip irdp preference 1000
```

Configure the multicast address 224.0.0.1 as the destination IP address for RAs sent by Vlan-interface 100.

```
[SwitchA-Vlan-interface100] ip irdp multicast
```

Specify the IP address 192.168.1.0 and preference 400 for Vlan-interface 100 to proxy-advertise.

```
[SwitchA-Vlan-interface100] ip irdp address 192.168.1.0 400
```

2. Configure Switch B

Specify the IP address of Vlan-interface 100.

```
<SwitchB> system-view
```

```
[SwitchB] interface Vlan-interface 100
```

```
[SwitchB-Vlan-interface100] ip address 10.154.5.2 24
```

Enable IRDP on Vlan-interface 100.

```
[SwitchB-Vlan-interface100] ip irdp
```

Specify preference 500 for the IP address of Vlan-interface 100.

```
[SwitchB-Vlan-interface100] ip irdp preference 500
```

Configure the multicast address 224.0.0.1 as the destination IP address for RAs sent by Vlan-interface 100.

```
[SwitchB-Vlan-interface100] ip irdp multicast
```

Specify the IP address 192.168.2.0 and preference 400 for Vlan-interface 100 to proxy-advertise.

```
[SwitchB-Vlan-interface100] ip irdp address 192.168.2.0 400
```

3. Verification

After enabling IRDP on Host A and Host B, which use Linux systems, display the routing table for the hosts (Host A for example).

```
[HostA@localhost ~]$ netstat -rne
```

Kernel IP routing table

| Destination | Gateway | Genmask | Flags | Metric | Ref | Use | Iface |
|-------------|------------|---------------|-------|--------|-----|-----|-------|
| 10.154.5.0 | 0.0.0.0 | 255.255.255.0 | U | 0 | 0 | 0 | eth1 |
| 192.168.1.0 | 0.0.0.0 | 255.255.255.0 | U | 0 | 0 | 0 | eth1 |
| 192.168.2.0 | 0.0.0.0 | 255.255.255.0 | U | 0 | 0 | 0 | eth1 |
| 0.0.0.0 | 10.154.5.1 | 0.0.0.0 | UG | 0 | 0 | 0 | eth1 |

The output shows that the default route on Host A points to IP address 10.154.5.1, and Host A has routes to 192.168.1.0/24 and 192.168.2.0/24.

UDP Helper configuration

UDP Helper can only be configured on VLAN interfaces and Layer 3 Ethernet interfaces.

Sometimes, a host must forward broadcasts to obtain network configuration information or request the names of other devices on the network. However, if the server or the device to be requested is located in another broadcast domain, the host cannot obtain such information through broadcast.

To solve this problem, the switch provides the UDP Helper function to relay specified UDP packets. UDP Helper functions as a relay agent that converts UDP broadcast packets into unicast packets and forwards them to a specified destination server.

With UDP Helper enabled, the switch decides whether to forward a received UDP broadcast packet according to the UDP destination port number of the packet.

- If the destination port number of the packet matches the one pre-configured on the switch, the switch modifies the destination IP address in the IP header, and then sends the packet to the specified destination server.
- If not, the switch sends the packet to the upper layer protocol for processing.

Configuring UDP Helper

| Step | Command | Remarks |
|--|--|---|
| 1. Enter system view. | system-view | — |
| 2. Enable UDP Helper. | udp-helper enable | Required. Disabled by default. |
| 3. Enable the forwarding of packets with the specified UDP destination port numbers. | udp-helper port { <i>port-number</i> dns netbios-ds netbios-ns tacacs tftp time } | Required. No UDP port number is specified by default. |
| 4. Enter interface view. | interface <i>interface-type</i> <i>interface-number</i> | — |
| 5. Specify the destination server to which UDP packets are to be forwarded. | udp-helper server <i>ip-address</i> | Required. No destination server is specified by default. |

The UDP Helper enabled device cannot forward DHCP broadcast packets. The UDP port number cannot be set to 67 or 68.

Specify a port number or the corresponding parameter for an UDP port to forward packets. For example, **udp-helper port 53** and **udp-helper port dns** specify the same UDP port number.

The configuration of all UDP ports is removed if you disable UDP Helper.

Configure up to 256 UDP port numbers to enable the forwarding of packets with these UDP port numbers.

Configure up to 20 destination servers on an interface.

Displaying and maintaining UDP Helper

| Task | Command | Remarks |
|--|---|------------------------|
| Displays the information of forwarded UDP packets. | display udp-helper server [interface <i>interface-type interface-number</i>] [[{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Clear statistics about packets forwarded. | reset udp-helper packet | Available in user view |

Configuration examples

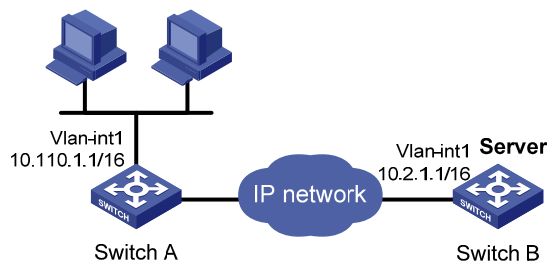
UDP Helper configuration example

Network requirements

As shown in [Figure 52](#), the IP address of VLAN-interface 1 of Switch A is 10.110.1.1/16, and the interface connects to the subnet 10.110.0.0/16.

Configure UDP Helper to forward broadcast packets with UDP destination port number 55 and destination IP address 255.255.255.255 or 10.110.255.255 to the destination server 10.2.1.1/16.

Figure 52 Network diagram for UDP Helper configuration



Configuration procedure

Make sure that a route from Switch A to the subnet 10.2.0.0/16 is available.

Enable Switch A to receive directed broadcasts.

```
<SwitchA> system-view
[SwitchA] ip forward-broadcast
```

Enable UDP Helper.

```
[SwitchA] udp-helper enable
```

Enable the forwarding broadcast packets with the UDP destination port 55.

```
[SwitchA] udp-helper port 55
```

Specify the destination server 10.2.1.1 on VLAN-interface 1.

```
[SwitchA] interface vlan-interface 1
[SwitchA-Vlan-interface1] ip address 10.110.1.1 16
[SwitchA-Vlan-interface1] udp-helper server 10.2.1.1
```

IPv6 basics configuration

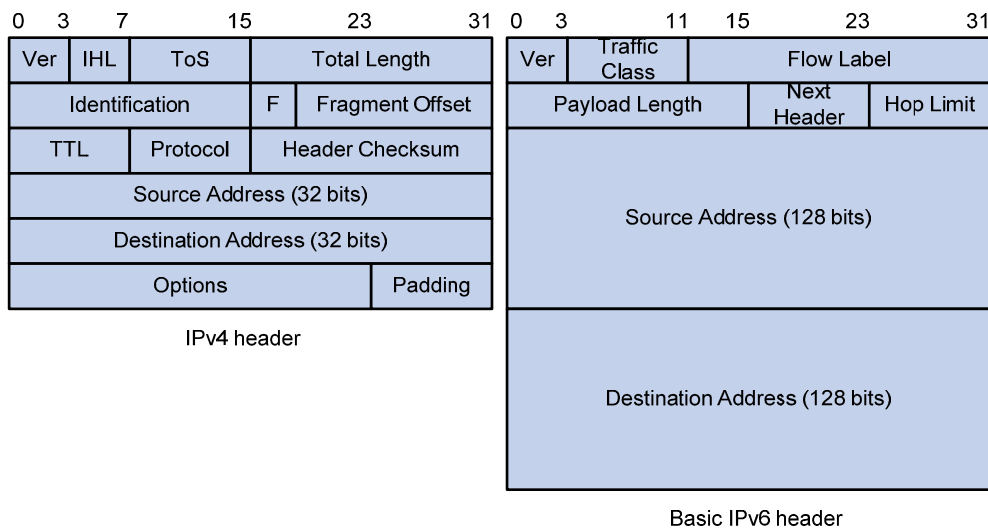
IPv6, also called IPng, was designed by the IETF as the successor to IPv4. The significant difference between IPv6 and IPv4 is that IPv6 increases the IP address size from 32 bits to 128 bits.

IPv6 features

Header format simplification

IPv6 removes several IPv4 header fields or moves them to the IPv6 extension headers to reduce the length of the basic IPv6 packet header. The basic IPv6 packet header has a fixed length of 40 bytes to simplify IPv6 packet handling and to improve the forwarding efficiency. Although an IPv6 address size is four times larger than an IPv4 address, the basic IPv6 packet header size is only twice the size of the option-less IPv4 packet header.

Figure 53 IPv4 packet header format and basic IPv6 packet header format



Larger address space

The source and destination IPv6 addresses are 128 bits (or 16 bytes) long. IPv6 can provide 3.4×10^{38} addresses to meet the requirements of hierarchical address division and the allocation of public and private addresses.

Hierarchical address structure

IPv6 uses the hierarchical address structure to quicken route searches faster and reduce the system sources occupied by the IPv6 routing table by route aggregation.

Address autoconfiguration

To simplify host configuration, IPv6 supports stateful and stateless address autoconfiguration.

- Stateful address autoconfiguration enables a host to acquire an IPv6 address and other configuration information from a server (for example, a DHCP server).

- Stateless address autoconfiguration enables a host to automatically generate an IPv6 address and other configuration information by using its link-layer address and the prefix information advertised by a router.

To communicate with other hosts on the same link, a host automatically generates a link-local address based on its link-layer address and the link-local address prefix (FE80::/10).

Built-in security

IPv6 defines extension headers to support IPsec. IPsec provides end-to-end security for network security solutions and enhances interoperability among different IPv6 applications.

QoS support

The Flow Label field in the IPv6 header allows the switch to label the packets and facilitates the special handling of a flow.

Enhanced neighbor discovery mechanism

The IPv6 neighbor discovery protocol is implemented through a group of ICMPv6 messages to manage the information exchange among neighboring nodes on the same link. The group of ICMPv6 messages replaces ARP messages, ICMPv4 Router Discovery messages, and ICMPv4 Redirect messages and provides a series of other functions.

Flexible extension headers

IPv6 cancels the Options field in the header and introduces optional extension headers to provide scalability and improve efficiency. The Options field in the IPv4 packet header contains 40 bytes at most, whereas the IPv6 extension headers are restricted only to the maximum size of IPv6 packets.

IPv6 addresses

IPv6 address format

CAUTION:

A double colon can appear once or not at all in an IPv6 address. Otherwise, the switch cannot determine how many zeros the double colons represent when converting them to zeros to restore a 128-bit IPv6 address.

An IPv6 address is represented as a set of 16-bit hexadecimal numbers separated by colons. An IPv6 address is divided into eight groups, and each 16-bit group is represented by four hexadecimal numbers, for example, 2001:0000:130F:0000:0000:09C0:876A:130B.

To simplify the representation of IPv6 addresses, handle zeros in IPv6 addresses by using the following methods.

- The leading zeros in each group can be removed. For example, the above address can be represented in a shorter format as 2001:0:130F:0:0:9C0:876A:130B.
- If an IPv6 address contains two or more consecutive groups of zeros, they can be replaced by a double colon (::). For example, the above address can be represented in the shortest format as 2001:0:130F::9C0:876A:130B.

An IPv6 address consists of an address prefix and an interface ID, which are equivalent to the network ID and the host ID of an IPv4 address respectively.

An IPv6 address prefix is written in IPv6-address/prefix-length notation where the IPv6-address is represented in any of the formats above and the prefix-length is a decimal number indicating how many leftmost bits of the IPv6 address comprises the address prefix.

IPv6 address types

IPv6 addresses fall into three types, unicast address, multicast address, and anycast address.

- **Unicast address**—An identifier for a single interface, similar to an IPv4 unicast address. A packet sent to a unicast address is delivered to the interface identified by that address.
- **Multicast address**—An identifier for a set of interfaces (typically belonging to different nodes), similar to an IPv4 multicast address. A packet sent to a multicast address is delivered to all interfaces identified by that address.
- **Anycast address**—An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to the nearest one of the interfaces identified by that address. The nearest interface is chosen according to the routing protocols' measure of distance.

There are no broadcast addresses in IPv6. Their function is replaced by multicast addresses.

The type of an IPv6 address is designated by the first several bits, the format prefix. [Table 7](#) lists the mappings between address types and format prefixes.

Table 7 Mappings between address types and format prefixes

| Type | Format prefix (binary) | IPv6 prefix ID | |
|-------------------|--|-------------------|-----------|
| Unicast address | Unspecified address | 00...0 (128 bits) | ::/128 |
| | Loopback address | 00...1 (128 bits) | ::1/128 |
| | Link-local address | 1111111010 | FE80::/10 |
| | Site-local address | 1111111011 | FEC0::/10 |
| | Global unicast address | Other forms | — |
| Multicast address | 11111111 | FF00::/8 | |
| Anycast address | Anycast addresses use the unicast address space and have the identical structure of unicast addresses. | | |

Unicast addresses

Unicast addresses comprise global unicast addresses, link-local unicast addresses, site-local unicast addresses, the loopback address, and the unspecified address.

- The global unicast addresses, equivalent to public IPv4 addresses, are provided for network service providers. This type of address allows efficient prefix aggregation to restrict the number of global routing entries.
- The link-local addresses are used for communication among link-local nodes for neighbor discovery and stateless autoconfiguration. Packets with link-local source or destination addresses are not forwarded to other links.
- The site-local unicast addresses are similar to private IPv4 addresses. Packets with site-local source or destination addresses are not forwarded out of the local site (or a private network).
- The loopback address is 0:0:0:0:0:0:1 (or ::1). It can never be assigned to any physical interface and can be used by a node to send an IPv6 packet to itself in the same way as the loopback address in IPv4.

- The unspecified address is 0:0:0:0:0:0:0:0 (or ::). It cannot be assigned to any node. Before acquiring a valid IPv6 address, a node fills this address in the source address field of IPv6 packets. The unspecified address cannot be used as a destination IPv6 address.

Multicast addresses

IPv6 multicast addresses listed in [Table 8](#) are reserved for special purposes.

Table 8 Reserved IPv6 multicast addresses

| Address | Application |
|---------|--|
| FF01::1 | Node-local scope all-nodes multicast address |
| FF02::1 | Link-local scope all-nodes multicast address |
| FF01::2 | Node-local scope all-routers multicast address |
| FF02::2 | Link-local scope all-routers multicast address |
| FF05::2 | Site-local scope all-routers multicast address |

Multicast addresses also include solicited-node addresses. A node uses a solicited-node multicast address to acquire the link-layer address of a neighboring node on the same link and to detect duplicate addresses. Each IPv6 unicast or anycast address has a corresponding solicited-node address. The format of a solicited-node multicast address is:

FF02:0:0:0:1:FFXX:XXXX

Where FF02:0:0:0:1:FF is fixed and consists of 104 bits, and XX:XXXX is the last 24 bits of an IPv6 unicast address or anycast address.

EUI-64 address-based interface identifiers

An interface identifier is 64 bits and uniquely identifies an interface on a link.

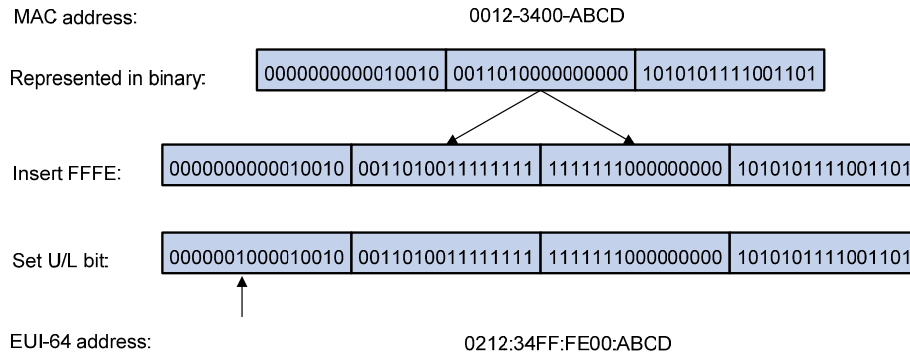
Interfaces generate EUI-64 address-based interface identifiers different.

- On an IEEE 802 interfaces (such as a VLAN interface)

The interface identifier is derived from the link-layer address (typically a MAC address) of the interface. To expand the 48-bit MAC address to a 64-bit interface identifier, the hexadecimal number FFFE (that is, 16 bits of 1111111111111110) is inserted into the MAC address (behind the 24th high-order bit). To make sure that the obtained interface identifier is globally unique, the U/L bit (which is the seventh high-order bit) is set to 1. Thus, an EUI-64 address-based interface identifier is obtained.

[Figure 54](#) shows the process of how an EUI-64 address-based interface identifier is generated from a MAC address.

Figure 54 Convert a MAC address into an EUI-64 address-based interface identifier



- On a tunnel interface

The lower 32 bits of the EUI-64 address-based interface identifier are the source IPv4 address of the tunnel interface. The higher 32 bits of the EUI-64 address-based interface identifier of an ISATAP tunnel interface are 0000:5EFE, whereas those of other tunnel interfaces are all zeros. For more information about tunnels, see the chapter “Tunneling configuration.”

- On an interface of another type

The EUI-64 address-based interface identifier is generated randomly by the switch.

IPv6 NDP

The IPv6 NDP uses five types of ICMPv6 messages to implement the following functions:

- [Address resolution](#)
- [Neighbor reachability detection](#)
- [Duplicate address detection](#)
- [Router/prefix discovery and address autoconfiguration](#)
- [Redirection](#)

Table 9 lists the types and functions of ICMPv6 messages used by the NDP.

Table 9 ICMPv6 messages used by ND

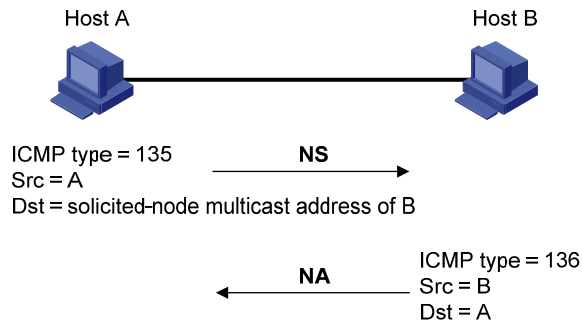
| ICMPv6 message | Type | Function |
|----------------|------|---|
| NS message | 135 | Acquires the link-layer address of a neighbor. |
| | | Verifies whether a neighbor is reachable. |
| NA message | 136 | Detects duplicate addresses. |
| | | Responds to an NS message. |
| RS message | 133 | Notifies the neighboring nodes of link layer changes. |
| | | Requests for an address prefix and other configuration information for autoconfiguration after startup. |
| RA message | 134 | Responds to an RS message. |
| | | Advertises information such as the Prefix Information options and flag bits. |

| ICMPv6 message | Type | Function |
|------------------|------|---|
| Redirect message | 137 | Informs the source host of a better next hop on the path to a particular destination when certain conditions are satisfied. |

Address resolution

Similar to the ARP function in IPv4, an IPv6 node acquires the link-layer addresses of neighboring nodes on the same link through NS and NA message exchanges. Figure 55 shows how Host A acquires the link-layer address of Host B on a single link.

Figure 55 Address resolution



The address resolution operates in the following steps.

1. Host A multicasts an NS message. The source address of the NS message is the IPv6 address of the sending interface of Host A and the destination address is the solicited-node multicast address of Host B. The NS message contains the link-layer address of Host A.
2. After receiving the NS message, Host B judges whether the destination address of the packet is its solicited-node multicast address. If yes, Host B learns the link-layer address of Host A, and then unicasts an NA message containing its link-layer address.
3. Host A acquires the link-layer address of Host B from the NA message.

Neighbor reachability detection

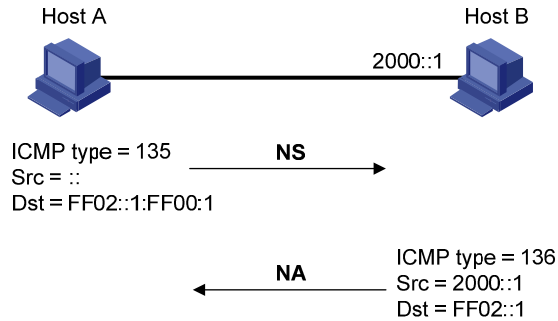
After Host A acquires the link-layer address of its neighbor Host B, Host A can use NS and NA messages to check whether Host B is reachable.

1. Host A sends an NS message whose destination address is the IPv6 address of Host B.
2. If Host A receives an NA message from Host B, Host A decides that Host B is reachable. Otherwise, Host B is unreachable.

Duplicate address detection

After Host A acquires an IPv6 address, it performs DAD to check whether the address is being used by any other node (similar to the gratuitous ARP function in IPv4). DAD is accomplished through NS and NA message exchanges. Figure 56 shows the DAD process.

Figure 56 Duplicate address detection



The DAD works in the following steps.

1. Host A sends an NS message whose source address is the unspecified address and whose destination address is the corresponding solicited-node multicast address of the IPv6 address to be detected. The NS message contains the IPv6 address.
2. If Host B uses this IPv6 address, Host B returns an NA message. The NA message contains the IPv6 address of Host B.
3. Host A learns that the IPv6 address is being used by Host B after receiving the NA message from Host B. If receiving no NA message, Host A decides that the IPv6 address is not in use and uses this address.

Router/prefix discovery and address autoconfiguration

Router/prefix discovery enables a node to locate the neighboring routers and to learn from the received RA message configuration parameters such as the prefix of the network where the node is located.

Stateless address autoconfiguration enables a node to generate an IPv6 address automatically according to the information obtained through router/prefix discovery.

Router/prefix discovery is implemented through RS and RA messages in the following steps:

1. At startup, a node sends an RS message to request from any available router for the address prefix and other configuration information for autoconfiguration.
2. A router returns an RA message containing information such as Prefix Information options. (The router also periodically sends an RA message.)
3. The node automatically generates an IPv6 address and other configuration information according to the address prefix and other configuration parameters in the RA message.

In addition to an address prefix, the Prefix Information option also contains the preferred lifetime and valid lifetime of the address prefix. Nodes update the preferred lifetime and valid lifetime accordingly through periodic RA messages.

An automatically generated address is applicable within the valid lifetime and is removed when the valid lifetime expires.

Redirection

A newly started host can contain only a default route to the gateway in its routing table. When certain conditions are satisfied, the gateway sends an ICMPv6 Redirect message to the source host so that the host can select a better next hop to forward packets (similar to the ICMP redirection function in IPv4).

The gateway sends an ICMPv6 Redirect message when the following conditions are satisfied.

- The receiving interface is the forwarding interface.

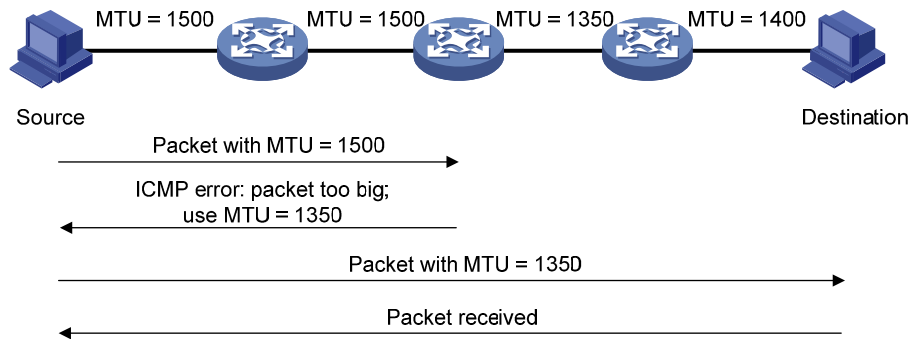
- The selected route itself is not created or modified by an ICMPv6 Redirect message.
- The selected route is not the default route.
- The IPv6 packet to be forwarded does not contain any routing header.

IPv6 PMTU discovery

The links that a packet passes from a source to a destination can have different MTUs. In IPv6, when the packet size exceeds the path MTU of a link, the packet is fragmented at the source end of the link to reduce the processing pressure on intermediate devices and use network resources effectively.

The PMTU discovery mechanism is to find the minimum MTU of all links in the path between a source and a destination. Figure 57 shows how a source host discovers the PMTU to a destination host.

Figure 57 PMTU discovery process



The PMTU discovery works in the following steps.

1. The source host compares its MTU with the packet to be sent, performs necessary fragmentation, and sends the resulting packet to the destination host.
2. If the MTU supported by a forwarding interface is smaller than the packet, the switch discards the packet and returns an ICMPv6 error packet containing the interface MTU to the source host.
3. After receiving the ICMPv6 error packet, the source host uses the returned MTU to limit the packet size, performs fragmentation, and sends the resulting packet to the destination host.
4. Step 2 and step 3 are repeated until the destination host receives the packet. In this way, the source host decides the minimum MTU of all links in the path to the destination host.

IPv6 transition technologies

Before IPv6 dominates the Internet, high-efficient, seamless IPv6 transition technologies are needed to enable communication between IPv4 and IPv6 networks. Several IPv6 transition technologies, which can be used in different environments and periods, such as dual stack (RFC 2893), tunneling (RFC 2893), and NAT-PT (RFC 2766).

- Dual stack is the most direct transition approach. A network node that supports both IPv4 and IPv6 is a dual stack node. A dual stack node configured with an IPv4 address and an IPv6 address can forward both IPv4 and IPv6 packets. For an upper layer application that supports both IPv4 and IPv6, either TCP or UDP can be selected at the transport layer, whereas the IPv6 stack is preferred at the network layer. Dual stack is suitable for communication between IPv4 nodes or between IPv6 nodes. It is the basis of all transition technologies. However, it does not solve the IPv4 address depletion issue because each dual stack node must have a globally unique IP address.

- Tunneling is an encapsulation technology that utilizes one network protocol to encapsulate packets of another network protocol and transfer them over the network.
- NAT-PT is usually applied on a switch between IPv4 and IPv6 networks to translate between IPv4 and IPv6 packets, allowing communication between IPv4 and IPv6 nodes. It performs IP address translation, and according to different protocols, performs semantic translation for packets. This technology is only suitable for communication between a pure IPv4 node and a pure IPv6 node.

The switch does not support NAT-PT.

Protocols and standards

Protocols and standards related to IPv6 include:

- RFC 1881, *IPv6 Address Allocation Management*
- RFC 1887, *An Architecture for IPv6 Unicast Address Allocation*
- RFC 1981, *Path MTU Discovery for IP version 6*
- RFC 2375, *IPv6 Multicast Address Assignments*
- RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 2461, *Neighbor Discovery for IP Version 6 (IPv6)*
- RFC 2462, *IPv6 Stateless Address Autoconfiguration*
- RFC 2463, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
- RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*
- RFC 2526, *Reserved IPv6 Subnet Anycast Addresses*
- RFC 3307, *Allocation Guidelines for IPv6 Multicast Addresses*
- RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture*

Configuration task list

| Task | Remarks | |
|----------------------------------|--|---------------------------|
| Configuring basic IPv6 functions | Enabling IPv6 | Required. |
| | Configuring an IPv6 global unicast address | Required to configure one |
| | Configuring an IPv6 link-local address | |
| | Configure an IPv6 anycast address | |
| Configuring IPv6 ND | Configuring a static neighbor entry | Optional. |
| | Configuring the maximum number of neighbors dynamically learned | Optional. |
| | Configuring RA message-related parameters | Optional. |
| | Configuring the maximum number of attempts to send an NS message for DAD | Optional. |
| | Setting the age timer for ND entries | Optional. |
| | Configuring ND snooping | Optional. |
| | Enabling ND proxy | Optional. |

| Task | | Remarks |
|-----------------------------------|--|-----------|
| Configuring PMTU discovery | Configuring a static PMTU for a specified IPv6 address | Optional. |
| | Configuring the aging time for dynamic PMTUs | Optional. |
| Configuring IPv6 TCP properties | | Optional. |
| Configuring ICMPv6 packet sending | Configuring the maximum ICMPv6 error packets sent in an interval | Optional. |
| | Enabling replying to multicast echo requests | Optional. |
| | Enabling sending of ICMPv6 time exceeded messages | Optional. |
| | Enabling sending of ICMPv6 destination unreachable messages | Optional. |

Configuring basic IPv6 functions

Enabling IPv6

Enable IPv6 before you perform any IPv6-related configuration. Without IPv6 enabled, an interface cannot forward IPv6 packets even if it has an IPv6 address configured.

To enable IPv6:

| Step | Command | Remarks |
|-----------------------|--------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable IPv6. | ipv6 | Required. Disabled by default. |

Configuring an IPv6 global unicast address

Configure an IPv6 global unicast address by using the following ways:

- **EUI-64 IPv6 addressing**—The IPv6 address prefix of an interface is manually configured, and the interface identifier is generated automatically by the interface.
- **Manual configuration**—The IPv6 global unicast address is configured manually.
- **Stateless address autoconfiguration**—The IPv6 global unicast address is generated automatically based on the address prefix information contained in the RA message.

Configure multiple IPv6 global unicast addresses with different prefixes on an interface.

A manually configured global unicast address takes precedence over an automatically generated one. If a global unicast address has been automatically generated on an interface when you manually configure another one with the same address prefix, the latter overwrites the previous. The overwritten automatic global unicast address is not restored even if the manual one is removed. Instead, a new global unicast address is automatically generated based on the address prefix information in the RA message that the interface receives at the next time.

Configuring an interface to generate EUI-64 IPv6 addresses

| Step | Command | Remarks |
|--|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure the interface to generate an EUI-64 IPv6 address. | ipv6 address ipv6-address/prefix-length eui-64 | Required. By default, no IPv6 global unicast address is configured on an interface. |

Specifying interface EUI-64 IPv6 address manually

| Step | Command | Remarks |
|--|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure an IPv6 address manually. | ipv6 address { ipv6-address prefix-length ipv6-address/prefix-length } | Required. By default, no IPv6 global unicast address is configured on an interface. |

Configuring an interface to generate IPv6 address using stateless address autoconfiguration

| Step | Command | Remarks |
|--|--|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view | interface interface-type interface-number | — |
| 3. Configure an IPv6 address to be generated through stateless address autoconfiguration | ipv6 address auto | Required. By default, no IPv6 global unicast address is configured on an interface. |

Using the **undo ipv6 address auto** command on an interface removes all IPv6 global unicast addresses automatically generated on the interface.

With stateless address autoconfiguration enabled on an interface, the switch automatically generates an IPv6 global unicast address by using the address prefix information in the received RA message and the interface ID. On an IEEE 802 interface (such as a VLAN interface), the interface ID is generated based on the MAC address of the interface, and is globally unique. As a result, the interface ID portion of the IPv6 global address remains unchanged and thus exposes the sender. An attacker can further exploit communication details such as the communication peer and time.

To fix the vulnerability, configure the temporary address function that enables the system to generate and use temporary IPv6 addresses with different interface ID portions on an interface. With this function configured on an IEEE 802 interface, the system can generate two addresses, public IPv6 address and temporary IPv6 address.

- **Public IPv6 address**—Comprises an address prefix provided by the RA message, and a fixed interface ID generated based on the MAC address of the interface.

- **Temporary IPv6 address**—Comprises an address prefix provided by the RA message, and a random interface ID generated through MD5.

Before sending a packet, the system preferably uses the temporary IPv6 address of the sending interface as the source address of the packet to be sent. When this temporary IPv6 address expires, the system removes it and generates a new one. This enables the system to send packets with different source addresses through the same interface. If the temporary IPv6 address cannot be used because of a DAD conflict, the public IPv6 address is used.

The preferred lifetime and valid lifetime for temporary IPv6 addresses are specified as follows:

- The preferred lifetime of a temporary IPv6 address takes the value of the smaller of the following values: the preferred lifetime of the address prefix in the RA message or the preferred lifetime configured for temporary IPv6 addresses minus DESYNC_FACTOR (which is a random number ranging 0 to 600, in seconds).
- The valid lifetime of a temporary IPv6 address takes the value of the smaller of the following values: the valid lifetime of the address prefix; valid lifetime configured for temporary IPv6 addresses.

To configure the temporary address function:

| Step | Command | Remarks |
|--|---|--|
| 1. Enter system view. | system-view | — |
| 2. Configure the system to generate and preferably use the temporary IPv6 address of the sending interface as the source address of the packet to be sent. | ipv6 prefer temporary-address [valid-lifetime preferred-lifetime] | Required. By default, the system does not generate or use any temporary IPv6 address. |

You must also enable stateless address autoconfiguration on an interface if you need temporary IPv6 addresses to be generated on that interface. Temporary IPv6 addresses do not override public IPv6 addresses. Therefore, an interface can have multiple IPv6 addresses with the same address prefix but different interface ID portions.

If the public IPv6 address fails to be generated on an interface due to a prefix conflict or other reasons, no temporary IPv6 address is generated on the interface.

Configuring an IPv6 link-local address

IPv6 link-local addresses can be configured in either of the following ways:

- **Automatic generation**—The switch automatically generates a link-local address for an interface according to the link-local address prefix (FE80::/10) and the link-layer address of the interface.
- **Manual assignment**—IPv6 link-local addresses can be assigned manually.

An interface can only have one link-local address. To avoid link-local address conflicts, use automatic generation method.

Manual assignment takes precedence over automatic generation. If you first use automatic generation and then manual assignment, the manually assigned link-local address overwrites the automatically generated one. If you first use manual assignment and then automatic generation, the automatically generated link-local address does not take effect and the link-local address is still the manually assigned one. If you delete the manually assigned address, the automatically generated link-local address is validated.

Configuring automatic generation of an IPv6 link-local address for an interface

| Step | Command | Remarks |
|--|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure the interface to automatically generate an IPv6 link-local address. | ipv6 address auto link-local | Optional. By default, no link-local address is configured on an interface. After an IPv6 global unicast address is configured on the interface, a link-local address is generated automatically. |

Configuring an IPv6 link-local address manually

| Step | Command | Remarks |
|--|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view | interface interface-type interface-number | — |
| 3. Configure an IPv6 link-local address manually | ipv6 address ipv6-address link-local | Optional. By default, no link-local address is configured on an interface. After an IPv6 global unicast address is configured on the interface, a link-local address is generated automatically. |

After an IPv6 global unicast address is configured for an interface, a link-local address is generated automatically. The automatically generated link-local address is the same as the one generated by using **ipv6 address auto link-local**. If a link-local address is manually assigned to an interface, this manual link-local address takes effect. If the manually assigned link-local address is removed, the automatically generated link-local address takes effect.

Undo ipv6 address auto link-local can only remove the link-local addresses generated through **ipv6 address auto link-local**. However, if an IPv6 global unicast address is already configured for an interface, the interface still has a link-local address because the system automatically generates one for the interface. If no IPv6 global unicast address is configured, the interface has no link-local address.

Configure an IPv6 anycast address

| Step | Command | Remarks |
|---------------------------------------|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure an IPv6 anycast address. | ipv6 address ipv6- address/ prefix-length anycast | Optional. By default, no IPv6 anycast address is configured on an interface. |

Configuring IPv6 NDP

Configuring a static neighbor entry entry

CAUTION:

Use either method above to configure a static neighbor entry for a VLAN interface.

- After a static neighbor entry is configured by using the first method, the switch must resolve the corresponding Layer 2 port information of the VLAN interface.
- If you use the second method, make sure that the corresponding VLAN interface exists and that the Layer 2 port specified by *port-type port-number* belongs to the VLAN specified by *vlan-id*. After a static neighbor entry is configured, the switch associates the VLAN interface with the IPv6 address to identify the static neighbor entry uniquely.

The IPv6 address of a neighboring node can be resolved into a link-layer address dynamically through NS and NA messages or through a manually configured static neighbor entry.

The switch uniquely identifies a static neighbor entry by the neighbor's IPv6 address and the local Layer 3 interface number. configure a static neighbor entry by using one of the following methods.

- Associate a neighbor IPv6 address and link-layer address with the Layer 3 interface of the local node.
- Associate a neighbor IPv6 address and link-layer address with a port in a VLAN containing the local node.

To configure a static neighbor entry:

| Step | Command | Remarks |
|---------------------------------------|---|-----------|
| 1. Enter system view. | system-view | — |
| 2. Configure a static neighbor entry. | ipv6 neighbor ipv6-address mac-address { <i>vlan-id port-type port-number</i> interface interface-type interface-number } [vpn-instance vpn-instance-name] | Required. |

Configuring the maximum number of neighbors dynamically learned

The switch can dynamically acquire the link-layer address of a neighboring node through NS and NA messages and add it into the neighbor table. A large table can reduce the forwarding performance of the switch. restrict the size of the neighbor table by setting the maximum number of neighbors that an interface can dynamically learn. When the number of dynamically learned neighbors reaches the threshold, the interface stops learning neighbor information.

To configure the maximum number of neighbors dynamically learned:

| Step | Command | Remarks |
|---|--|---|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure the maximum number of neighbors dynamically learned by an interface. | ipv6 neighbors max-learning-number | Optional. By default, an interface on an HP 5800 switch can learn up to 8192 neighbors dynamically; an interface on an HP 5820X switch can learn up to 4096 neighbors dynamically. |

Configuring RA message-related parameters

Enable an interface to send RA messages, and configure the interval for sending RA messages and parameters in RA messages. After receiving an RA message, a host can use these parameters to perform corresponding operations. Table 10 lists the configurable parameters in an RA message and their descriptions.

Table 10 Parameters in an RA message and their descriptions

| Parameters | Description |
|----------------------------|---|
| Cur Hop Limit | When sending an IPv6 packet, a host uses the value to fill the Hop Limit field in IPv6 headers. The value is also filled into the Hop Limit field in the response packet of a switch. |
| Prefix Information options | After receiving the prefix information advertised by the switch, the hosts on the same link can perform stateless autoconfiguration. |
| MTU | Ensures that all nodes on a link use the same MTU value. |
| M flag | Determines whether hosts use the stateful autoconfiguration to acquire IPv6 addresses. If the M flag is set to 1, hosts use the stateful autoconfiguration (for example, through a DHCP server) to acquire IPv6 addresses. Otherwise, hosts use the stateless autoconfiguration to acquire IPv6 addresses and generate IPv6 addresses according to their own link-layer addresses and the prefix information advertised by the router. |

| Parameters | Description |
|-----------------|--|
| O flag | Determines whether hosts use stateful autoconfiguration to acquire other configuration information. If the O flag is set to 1, hosts use stateful autoconfiguration (for example, through a DHCP server) to acquire other configuration information. Otherwise, hosts use stateless autoconfiguration to acquire other configuration information. |
| Router Lifetime | This field tells the receiving hosts how long this router can serve as a default router. According to the router lifetime in the received RA messages, hosts determine whether the router sending RA messages can serve as the default router. |
| Retrans Timer | If the switch fails to receive a response message within the specified time after sending an NS message, the switch retransmits the NS message. |
| Reachable Time | If the neighbor reachability detection shows that a neighbor is reachable, the switch considers the neighbor reachable within the specified reachable time. If the switch must send a packet to a neighbor after the specified reachable time expires, the switch reconfirms whether the neighbor is reachable. |

Allowing the sending of RA messages

| Step | Command | Remarks |
|---|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Disable RA message suppression. | undo ipv6 nd ra halt | Required. By default, RA messages are suppressed. |
| 4. Configure the maximum and minimum intervals for sending RA messages. | ipv6 nd ra interval max-interval-value min-interval-value | Optional. By default, the maximum interval for sending RA messages is 600 seconds, and the minimum interval is 200 seconds. The switch sends RA messages at random intervals between the maximum interval and the minimum interval. The minimum interval should be less than or equal to 0.75 times the maximum interval. |

Configuring parameters related to RA messages

| Step | Command | Remarks |
|---|---|---|
| 1. Enter system view. | system-view | — |
| 2. Configure the hop limit. | ipv6 nd hop-limit <i>value</i> | Optional. 64 by default. |
| 3. Enter interface view. | interface <i>interface-type</i> <i>interface-number</i> | — |
| 4. Configure the prefix information in RA messages. | ipv6 nd ra prefix { <i>ipv6-prefix</i> <i>prefix-length</i> <i>ipv6-prefix</i> / <i>prefix-length</i> } <i>valid-lifetime</i> <i>preferred-lifetime</i> [no-autoconfig off-link] * | Optional. By default, no prefix information is configured for RA messages, and the IPv6 address of the interface sending RA messages is used as the prefix information with valid lifetime 2592000 seconds (that is, 30 days) and preferred lifetime 604800 seconds (that is, 7 days). |
| 5. Turn off the MTU option in RA messages. | ipv6 nd ra no-advlinkmtu | Optional. By default, RA messages contain the MTU option. |
| 6. Set the M flag bit to 1. | ipv6 nd autoconfig managed-address-flag | Optional. By default, the M flag bit is set to 0, and hosts acquire IPv6 addresses through stateless autoconfiguration. |
| 7. Set the O flag bit to 1. | ipv6 nd autoconfig other-flag | Optional. By default, the O flag bit is set to 0, and hosts acquire other configuration information through stateless autoconfiguration. |
| 8. Configure the router lifetime in RA messages. | ipv6 nd ra router-lifetime <i>value</i> | Optional. 1800 seconds by default. |
| 9. Set the NS retransmission timer. | ipv6 nd ns retrans-timer <i>value</i> | Optional. By default, the local interface sends NS messages at 1000 millisecond intervals, and the value of the Retrans Timer field in RA messages sent by the local interface is 0. The interval for retransmitting an NS message is determined by the receiving switch. |
| 10. Set the reachable time. | ipv6 nd nud reachable-time <i>value</i> | Optional. By default, the neighbor reachable time on the local interface is 30000 milliseconds, and the value of the Reachable Time field in the RA messages sent by the local interface is 0. The neighbor reachable time is determined by the receiving switch. |

The maximum interval for sending RA messages should be less than or equal to the router lifetime in RA messages, so that the router can be updated through an RA message before expiration.

The values of the NS retransmission timer and the reachable time configured for an interface are sent to hosts via RA messages. Furthermore, this interface sends NS messages at the interval of the NS retransmission timer and considers a neighbor reachable within the reachable time.

Configuring the maximum number of attempts to send an NS message for DAD

An interface sends an NS message for DAD after acquiring an IPv6 address. If the interface does not receive a response within a specified time (determined by **ipv6 nd ns retrans-timer**), it continues to send an NS message. If it still does not receive a response after the number of sent attempts reaches the threshold (specified with **ipv6 nd dad attempts**), the acquired address is considered usable.

To configure the attempts to send an NS message for DAD:

| Step | Command | Remarks |
|--|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure the number of attempts to send an NS message for DAD. | ipv6 nd dad attempts value | Optional. 1 by default. When the value argument is set to 0, DAD is disabled. |

Setting the age timer for ND entries

ND entries have an age timer. If an ND entry is not refreshed within a certain time after aging out, the switch sends an NS message for detection. If no response is received, it removes the ND entry. set the age timer as needed.

To set the age timer for ND entries:

| Step | Command | Remarks |
|--------------------------------------|---|-------------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Set the age timer for ND entries. | ipv6 neighbor stale-aging aging-time | Optional. Four hours by default. |

Configuring ND snooping

The ND snooping feature is used in Layer 2 switching networks. It creates ND snooping entries using DAD NS messages.

ND snooping entries are used to:

- Cooperate with the ND detection function. For more information about ND detection, see the *Security Configuration Guide*.
- Cooperate with the IP Source Guard function. For more information about IP source guard, see the *Security Configuration Guide*.

After you enable ND snooping on a VLAN of a switch, ND packets received by the interfaces of the VLAN are redirected to the CPU. The CPU uses the ND packets to create or update ND snooping entries comprising source IPv6 address, source MAC address, receiving VLAN, and receiving port information.

The following items describe how an ND snooping entry is created, updated, and aged out.

1. Creating an ND snooping entry

The switch only uses received DAD NS messages to create ND snooping entries.

2. Updating an ND snooping entry

Upon receiving an ND packet, the switch searches the ND snooping table for an entry containing the source IPv6 address and VLAN of the packet, and then matches the MAC address of the ND packet against that in the entry.

- If the MAC address of the ND packet matches that in the entry, the switch updates the receiving port and aging time of the ND snooping entry if the receiving ports are different, or only the aging time of the entry if the receiving ports are the same).
- If the MAC addresses do not match and the received packet is a DAD NS message, the message is ignored.
- If the MAC addresses do not match and the received packet is not a DAD NS message, the switch performs active acknowledgement.

The active acknowledgement is performed in the following steps:

- The switch checks the validity of the existing ND snooping entry. The switch sends out a DAD NS message including the IPv6 address of the ND snooping entry every one second for three times at most. If a corresponding NA message (whose source IPv6 address, source MAC address, and source VLAN are consistent with those of the existing entry) is received, the switch stops sending DAD NS messages and updates the receiving port and aging time of the existing entry if the receiving ports are different, or only the aging time of the entry if the receiving ports are the same. If no corresponding NA message is received within five seconds after the first DAD NS message is sent, the switch starts to check the validity of the received ND packet.
- To check the validity of the received ND packet (packet A for example), the switch sends out a DAD NS message including the source IPv6 address of packet A every one second for three times at most. If a corresponding NA message (whose source IPv6 address, source MAC address, and source VLAN are consistent with those of packet A) is received, the switch stops sending DAD NS messages and updates the receiving port and aging time of the entry if the receiving ports are different, or only the aging time of the entry if the receiving ports are the same. If no corresponding NA message is received within five seconds after the first DAD NS message is sent, the switch does not update the entry.

3. Aging out an ND snooping entry

An ND snooping entry is aged out after 25 minutes. If an ND snooping entry is not updated within 15 minutes, the switch performs active acknowledgement.

The switch sends out a DAD NS message including the IPv6 address of the ND snooping entry every one second for three times at most.

- If a corresponding NA message is received (the source IPv6 address, source MAC address, and source VLAN are consistent with those of the existing entry), the switch stops sending DAD NS messages and updates receiving port and aging time of the existing entry if the receiving ports are different, or only the aging time if the receiving ports are the same.
- If no corresponding NA message is received within five seconds after the first DAD NS message is sent out, the switch removes the entry when the timer expires.

Configuration procedure

| Step | Command | Remarks |
|---|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter VLAN view. | vlan vlan-id | — |
| 3. Enable ND snooping. | ipv6 nd snooping enable | Required. Disabled by default. |
| 4. Return to system view. | quit | — |
| 5. Enter Layer 2 Ethernet interface view/Layer 2 aggregate interface view. | interface interface-type interface-number | — |
| 6. Configure the maximum number of ND snooping entries the interface can learn. | ipv6 nd snooping max-learning-num number | Optional. By default, the number of ND snooping entries an interface can learn is unlimited. |

Enabling ND proxy

ND proxy supports only the NS and NA messages.

ND proxy forwards NS and NA messages between hosts on the same subnet but different broadcast domains.

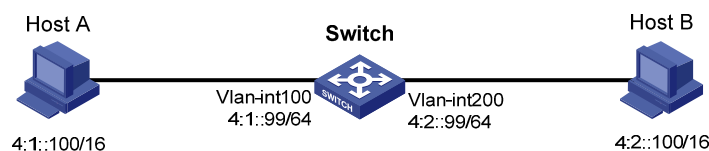
ND proxy falls into common ND proxy and local ND proxy.

Unless otherwise specified, ND proxy described in the following text refers to common ND proxy.

- ND proxy

As shown in [Figure 58](#), VLAN-interface 100 with IPv6 address 4:1::99/64 and VLAN-interface 200 with IP address 4:2::99/64 belong to different subnets. Host A and Host B reside on the same network but in different broadcast domains.

Figure 58 ND proxy network diagram



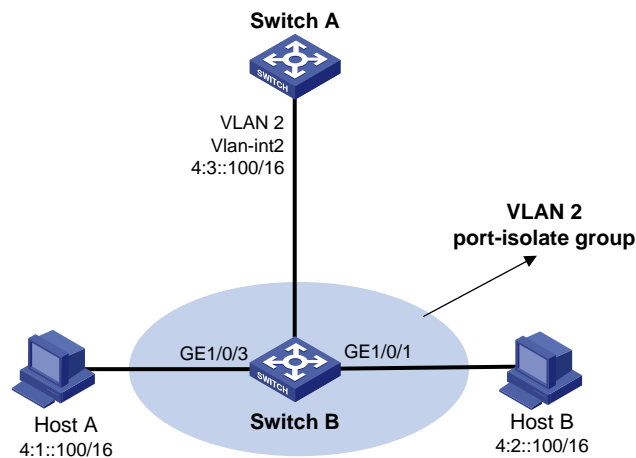
Because Host A's IPv6 address is on the same subnet as Host B's, Host A directly sends an NS to obtain Host B's MAC address. However, Host B cannot receive the NS message because they belong to different broadcast domains.

To solve this problem, enable ND proxy on VLAN-interface 100 and VLAN-interface 200 of the switch. The switch finds the matching forwarding entry according to the destination IPv6 address of the NS message and sends the message through the output interface of that entry. Upon receiving the NS, Host B sends an NA message to the switch, which forwards it to Host A.

- Local ND proxy

As shown in Figure 59, both Host A and Host B belong to VLAN 2, but they connect to GigabitEthernet 1/0/3 and GigabitEthernet 1/0/1 respectively, which are isolated at Layer 2.

Figure 59 Local ND proxy network diagram



Because Host A's IPv6 address is on the same subnet as Host B's, Host A directly sends an NS to obtain Host B's MAC address. However, Host B cannot receive the NS because they are isolated at Layer 2.

To solve this problem, enable local ND proxy on VLAN-interface 2 of Switch A so that Switch A can forward messages between Host A and Host B.

Enabling ND proxy

| Step | Command | Remarks |
|---|---|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enter VLAN interface or Layer 3 Ethernet interface view. | interface interface-type interface-number | — |
| 3. Enable ND proxy. | proxy-nd enable | Optional. Disabled by default. |

Enabling local ND proxy

| Step | Command | Remarks |
|---|---|---------|
| 1. Enter system view. | system-view | — |
| 2. Enter VLAN interface or Layer 3 Ethernet interface view. | interface interface-type interface-number | — |

| Step | Command | Remarks |
|---------------------------|------------------------------|-----------------------------------|
| 3. Enable local ND proxy. | local-proxy-nd enable | Optional. Disabled by default. |

Configuring PMTU discovery

Configuring a static PMTU for a specified IPv6 address

Configure a static PMTU for a specified destination IPv6 address. When a source host sends a packet through an interface, it compares the interface MTU with the static PMTU of the specified destination IPv6 address. If the packet size is larger than the smaller one between the two values, the host fragments the packet according to the smaller value.

To configure a static PMTU for a specified IPv6 address:

| Step | Command | Remarks |
|--|---|--|
| 1. Enter system view. | system-view | — |
| 2. Configure a static PMTU for a specified IPv6 address. | ipv6 pathmtu [vpn-instance vpn-instance-name] ipv6-address [value] | Required. By default, no static PMTU is configured. |

Configuring the aging time for dynamic PMTUs

After the path MTU from a source host to a destination host is dynamically determined (see “[IPv6 PMTU discovery](#)”), the source host sends subsequent packets to the destination host based on this MTU. After the aging time expires, the dynamic PMTU is removed and the source host re-determines a dynamic path MTU through the PMTU mechanism.

The aging time is invalid for a static PMTU.

To configure the aging time for dynamic PMTUs:

| Step | Command | Remarks |
|--|----------------------------------|-------------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Configure the aging time for dynamic PMTUs. | ipv6 pathmtu age age-time | Optional. 10 minutes by default. |

Configuring IPv6 TCP properties

Configure the following IPv6 TCP properties.

- **Synwait timer**—When a SYN packet is sent, the synwait timer is triggered. If no response packet is received before the synwait timer expires, the IPv6 TCP connection establishment fails.
- **Finwait timer**—When the IPv6 TCP connection status is FIN_WAIT_2, the finwait timer is triggered. If no packet is received before the finwait timer expires, the IPv6 TCP connection is terminated. If a FIN packet is received, the IPv6 TCP connection status becomes TIME_WAIT. If non-FIN packets are received, the finwait timer is reset upon receipt of the last non-FIN packet and the connection is terminated after the finwait timer expires.
- Size of the IPv6 TCP sending/receiving buffer.

To configure IPv6 TCP properties:

| Step | Command | Remarks |
|---|--|--------------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Set the synwait timer. | tcp ipv6 timer syn-timeout <i>wait-time</i> | Optional. 75 seconds by default. |
| 3. Set the finwait timer. | tcp ipv6 timer fin-timeout <i>wait-time</i> | Optional. 675 seconds by default. |
| 4. Set the size of the IPv6 TCP sending/receiving buffer. | tcp ipv6 window <i>size</i> | Optional. 8 KB by default. |

Configuring ICMPv6 packet sending

Configuring the maximum ICMPv6 error packets sent in an interval

If too many ICMPv6 error packets are sent within a short time in a network, network congestion can occur. To avoid network congestion, control the maximum number of ICMPv6 error packets sent within a specified time by adopting the token bucket algorithm.

set the capacity of a token bucket to determine the number of tokens in the bucket. In addition, set the update interval of the token bucket, that is, the interval for restoring the configured capacity. One token allows one ICMPv6 error packet to be sent. Each time an ICMPv6 error packet is sent, the number of tokens in a token bucket decreases by one. If the number of ICMPv6 error packets successively sent exceeds the capacity of the token bucket, the additional ICMPv6 error packets cannot be sent out until the capacity of the token bucket is restored.

To configure the capacity and update interval of the token bucket:

| Step | Command | Remarks |
|--|--|--|
| 1. Enter system view. | system-view | — |
| 2. Configure the capacity and update interval of the token bucket. | ipv6 icmp-error { bucket bucket-size ratelimit interval } * | Optional. By default, the capacity of a token bucket is 10 and the update interval is 100 milliseconds. At most 10 ICMPv6 error packets can be sent within 100 milliseconds. The update interval "0" indicates that the number of ICMPv6 error packets sent is not restricted. |

Enabling replying to multicast echo requests

If hosts are configured to answer multicast echo requests, an attacker can use this mechanism to attack a host. For example, if Host A (an attacker) sends an echo request with the source being Host B to a multicast address, all hosts in the multicast group sends echo replies to Host B. To prevent such an attack, disable the switch from replying multicast echo requests by default. In some application scenarios, however, you need to enable the switch to reply multicast echo requests.

To enable replying to multicast echo requests:

| Step | Command | Remarks |
|--|--|--------------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable replying to multicast echo requests. | ipv6 icmpv6 multicast-echo-reply enable | Required. Not enabled by default. |

Enabling sending of ICMPv6 time exceeded messages

A switch sends out an ICMPv6 Time Exceeded message in the following cases.

- If a received IPv6 packet's destination IP address is not a local address and its hop limit is 1, the switch sends an ICMPv6 Hop Limit Exceeded message to the source.
- Upon receiving the first fragment of an IPv6 datagram with the destination IP address being the local address, the switch starts a timer. If the timer expires before all fragments arrive, an ICMPv6 Fragment Reassembly Timeout message is sent to the source.

If large amounts of malicious packets are received, the performance of the switch degrades greatly because it has to send back ICMP Time Exceeded messages. disable sending of ICMPv6 Time Exceeded messages.

To enable sending of ICMPv6 time exceeded messages:

| Step | Command | Remarks |
|---|-------------------------------------|----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable sending of ICMPv6 Time Exceeded messages. | ipv6 hoplimit-expires enable | Optional. Enabled by default. |

Enabling sending of ICMPv6 destination unreachable messages

If the switch fails to forward a received IPv6 packet due to one of the following reasons, it drops the packet and sends a corresponding ICMPv6 Destination Unreachable error message to the source.

- If no route is available for forwarding the packet, the switch sends a "no route to destination" ICMPv6 error message to the source.
- If the switch fails to forward the packet due to administrative prohibition (such as a firewall filter or an ACL), the switch sends the source a "destination network administratively prohibited" ICMPv6 error message.
- If the switch fails to deliver the packet because the destination is beyond the scope of the source IPv6 address (for example, the source IPv6 address of the packet is a link-local address whereas the destination IPv6 address of the packet is a global unicast address), the switch sends the source a "beyond scope of source address" ICMPv6 error message.
- If the switch fails to resolve the corresponding link layer address of the destination IPv6 address, the switch sends the source an "address unreachable" ICMPv6 error message.
- If the packet with the destination being local and transport layer protocol being UDP and the packet's destination port number does not match the running process, the switch sends the source a "port unreachable" ICMPv6 error message.

If an attacker sends abnormal traffic that causes the switch to generate ICMPv6 destination unreachable messages, end users can be affected. To prevent such attacks, disable the switch from sending ICMPv6 destination unreachable messages.

To enable sending of ICMPv6 destination unreachable messages:

| Step | Command | Remarks |
|---|--------------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable sending of ICMPv6 destination unreachable messages. | ipv6 unreachable enable | Required. Disabled by default. |

Displaying and maintaining IPv6 basics configuration

| Task | Command | Remarks |
|---|---|-----------------------|
| Display IPv6 FIB entries. | display ipv6 fib [vpn-instance <i>vpn-instance-name</i>] [acl6 <i>acl6-number</i> ipv6-prefix <i>ipv6-prefix-name</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display the IPv6 FIB entry of a specified destination IPv6 address. | display ipv6 fib [vpn-instance <i>vpn-instance-name</i>] <i>ipv6-address</i> [<i>prefix-length</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display the IPv6 information of the interface. | display ipv6 interface [<i>interface-type</i> [<i>interface-number</i>]] [verbose] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |

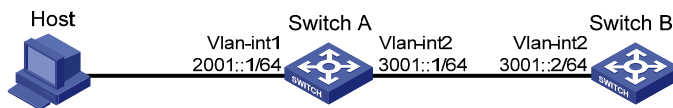
| Task | Command | Remarks |
|---|--|------------------------|
| Display neighbor information. | display ipv6 neighbors { { <i>ipv6-address</i> all dynamic static } [<i>slot slot-number</i>] interface <i>interface-type interface-number</i> vlan <i>vlan-id</i> } [verbose] [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display the total number of neighbor entries satisfying the specified conditions. | display ipv6 neighbors { { all dynamic static } [<i>slot slot-number</i>] interface <i>interface-type interface-number</i> vlan <i>vlan-id</i> } count [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display the neighbor information of VPN instances. | display ipv6 neighbors vpn-instance <i>vpn-instance-name</i> [count] [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display the IPv6 PMTU information. | display ipv6 pathmtu [vpn-instance <i>vpn-instance-name</i>] { { <i>ipv6-address</i> all dynamic static } [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display socket information. | display ipv6 socket [socket-type <i>socket-type</i>] [<i>task-id socket-id</i>] [<i>slot slot-number</i>] [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display the statistics of IPv6 packets and ICMPv6 packets. | display ipv6 statistics [<i>slot slot-number</i>] [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display the IPv6 TCP connection statistics. | display tcp ipv6 statistics [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display the IPv6 TCP connection status information. | display tcp ipv6 status [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display the IPv6 UDP connection statistics. | display udp ipv6 statistics [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Display ND snooping entries. | display ipv6 nd snooping [<i>ipv6-address</i> vlan <i>vlan-id</i>] [[{ begin exclude include } <i>regular-expression</i>]] | Available in any view |
| Clear IPv6 neighbor information. | reset ipv6 neighbors { all dynamic interface <i>interface-type interface-number</i> slot <i>slot-number</i> static } | Available in user view |
| Clear the PMTU values. | reset ipv6 pathmtu { all static dynamic } | Available in user view |
| Clear the statistics of IPv6 and ICMPv6 packets. | reset ipv6 statistics [<i>slot slot-number</i>] | Available in user view |
| Clear all IPv6 TCP connection statistics. | reset tcp ipv6 statistics | Available in user view |
| Clear the statistics of all IPv6 UDP packets. | reset udp ipv6 statistics | Available in user view |
| Clear ND snooping entries. | reset ipv6 nd snooping [<i>ipv6-address</i> vlan <i>vlan-id</i>] | Available in user view |

IPv6 configuration example

Network requirements

- As shown in [Figure 60](#), a host, Switch A and Switch B are connected through Ethernet ports. Add the Ethernet ports into corresponding VLANs, configure IPv6 addresses for the VLAN interfaces and verify that they are connected.
- The global unicast addresses of VLAN-interface 1 and VLAN-interface 2 on Switch A are 2001::1/64 and 3001::1/64 respectively.
- The global unicast address of VLAN-interface 2 on Switch B is 3001::2/64, and a route to Host is available.
- IPv6 is enabled for Host to automatically obtain an IPv6 address through IPv6 ND, and a route to Switch B is available.

Figure 60 Network diagram for IPv6 address configuration (on switches)



The VLAN interfaces have been created on the switch.

Configuration procedure

1. Configure Switch A

Enable IPv6.

```
<SwitchA> system-view  
[SwitchA] ipv6
```

Specify a global unicast address for VLAN-interface 2.

```
[SwitchA] interface vlan-interface 2  
[SwitchA-Vlan-interface2] ipv6 address 3001::1/64  
[SwitchA-Vlan-interface2] quit
```

Specify a global unicast address for VLAN-interface 1, and allow it to advertise RA messages (no interface advertises RA messages by default).

```
[SwitchA] interface vlan-interface 1  
[SwitchA-Vlan-interface1] ipv6 address 2001::1/64  
[SwitchA-Vlan-interface1] undo ipv6 nd ra halt  
[SwitchA-Vlan-interface1] quit
```

2. Configure Switch B

Enable IPv6.

```
<SwitchB> system-view  
[SwitchB] ipv6
```

Configure a global unicast address for VLAN-interface 2.

```
[SwitchB] interface vlan-interface 2  
[SwitchB-Vlan-interface2] ipv6 address 3001::2/64  
[SwitchB-Vlan-interface2] quit
```



```
# Configure an IPv6 static route with destination IP address 2001::/64 and next hop address 3001::1.
[SwitchB] ipv6 route-static 2001:: 64 3001::1
```

3. Configure Host

```
# Enable IPv6 for Host to automatically obtain an IPv6 address through IPv6 ND.
```

```
# Display the neighbor information of GigabitEthernet 1/0/2 on Switch A.
```

```
[SwitchA] display ipv6 neighbors interface GigabitEthernet 1/0/2
                Type: S-Static    D-Dynamic
IPv6 Address          Link-layer      VID  Interface    State T Age
FE80::215:E9FF:FEA6:7D14  0015-e9a6-7d14  1    GE1/0/2      STALE D 1238
2001::15B:E0EA:3524:E791  0015-e9a6-7d14  1    GE1/0/2      STALE D 1248
```

The information shows that the IPv6 global unicast address that Host obtained is 2001::15B:E0EA:3524:E791.

Verification

CAUTION:

When you ping a link-local address, you should use the `-i` parameter to specify an interface for the link-local address.

```
# Display the IPv6 interface settings on Switch A. All the IPv6 global unicast addresses configured on the interface are displayed.
```

```
[SwitchA] display ipv6 interface vlan-interface 2 verbose
Vlan-interface2 current state :UP
Line protocol current state :UP
IPv6 is enabled, link-local address is FE80::20F:E2FF:FE00:2
  Global unicast address(es):
    3001::1, subnet is 3001::/64
  Joined group address(es):
    FF02::1:FF00:0
    FF02::1:FF00:1
    FF02::1:FF00:2
    FF02::2
    FF02::1
  MTU is 1500 bytes
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
  InReceives:                25829
  InTooShorts:                0
  InTruncatedPkts:           0
  InHopLimitExceeds:         0
  InBadHeaders:               0
  InBadOptions:               0
  ReasmReqds:                 0
  ReasmOKs:                   0
```

```

InFragDrops:          0
InFragTimeouts:      0
OutFragFails:        0
InUnknownProtos:    0
InDelivers:          47
OutRequests:         89
OutForwDatagrams:   48
InNoRoutes:          0
InTooBigErrors:     0
OutFragOKs:          0
OutFragCreates:     0
InMcastPkts:        6
InMcastNotMembers: 25747
OutMcastPkts:       48
InAddrErrors:        0
InDiscards:          0
OutDiscards:         0

```

```

[SwitchA] display ipv6 interface vlan-interface 1 verbose
Vlan-interfacel current state :UP
Line protocol current state :UP
IPv6 is enabled, link-local address is FE80::20F:E2FF:FE00:1C0
Global unicast address(es):
  2001::1, subnet is 2001::/64
Joined group address(es):
  FF02::1:FF00:0
  FF02::1:FF00:1
  FF02::1:FF00:1C0
  FF02::2
  FF02::1
MTU is 1500 bytes
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 600 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
InReceives:          272
InTooShorts:         0
InTruncatedPkts:    0
InHopLimitExceeds:  0
InBadHeaders:        0
InBadOptions:        0
ReasmReqds:          0
ReasmOKs:            0

```

```

InFragDrops:          0
InFragTimeouts:      0
OutFragFails:        0
InUnknownProtos:    0
InDelivers:          159
OutRequests:         1012
OutForwDatagrams:   35
InNoRoutes:          0
InTooBigErrors:     0
OutFragOKs:          0
OutFragCreates:     0
InMcastPkts:        79
InMcastNotMembers:  65
OutMcastPkts:       938
InAddrErrors:        0
InDiscards:          0
OutDiscards:         0

```

Display the IPv6 interface settings on Switch B. All the IPv6 global unicast addresses configured on the interface are displayed.

```

[SwitchB] display ipv6 interface vlan-interface 2 verbose
Vlan-interface2 current state :UP
Line protocol current state :UP
IPv6 is enabled, link-local address is FE80::20F:E2FF:FE00:1234
Global unicast address(es):
  3001::2, subnet is 3001::/64
Joined group address(es):
  FF02::1:FF00:0
  FF02::1:FF00:2
  FF02::1:FF00:1234
  FF02::2
  FF02::1
MTU is 1500 bytes
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND retransmit interval is 1000 milliseconds
Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
InReceives:          117
InTooShorts:         0
InTruncatedPkts:    0
InHopLimitExceeds:  0
InBadHeaders:        0
InBadOptions:        0
ReasmReqds:          0
ReasmOKs:            0
InFragDrops:         0
InFragTimeouts:     0

```

```

OutFragFails:          0
InUnknownProtos:      0
InDelivers:           117
OutRequests:          83
OutForwDatagrams:     0
InNoRoutes:           0
InTooBigErrors:       0
OutFragOKs:           0
OutFragCreates:       0
InMcastPkts:          28
InMcastNotMembers:    0
OutMcastPkts:         7
InAddrErrors:         0
InDiscards:           0
OutDiscards:          0

```

Ping Switch A and Switch B on Host, and ping Switch A and Host on Switch B to verify that they are connected.

```

[SwitchB] ping ipv6 -c 1 3001::1
PING 3001::1 : 56 data bytes, press CTRL_C to break
  Reply from 3001::1
    bytes=56 Sequence=1 hop limit=64 time = 2 ms

--- 3001::1 ping statistics ---
  1 packet(s) transmitted
  1 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/2/2 ms
[SwitchB-Vlan-interface2] ping ipv6 -c 1 2001::15B:E0EA:3524:E791
PING 2001::15B:E0EA:3524:E791 : 56 data bytes, press CTRL_C to break
  Reply from 2001::15B:E0EA:3524:E791
    bytes=56 Sequence=1 hop limit=63 time = 3 ms

--- 2001::15B:E0EA:3524:E791 ping statistics ---
  1 packet(s) transmitted
  1 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 3/3/3 ms

```

As shown in the output information, Switch B can ping Switch A and Host.

Troubleshooting IPv6 basics configuration

Symptom

The peer IPv6 address cannot be pinged.

Solution

- Use **display current-configuration** in any view or **display this** in system view to verify that IPv6 is enabled.
- Use **display ipv6 interface** in any view to verify that the IPv6 address of the interface is correct and the interface is up.
- Use **debugging ipv6 packet** in user view to enable the debugging for IPv6 packets to help locate the cause.

DHCPv6 overview

The DHCPv6 was designed based on IPv6 addressing scheme and is used for assigning IPv6 prefixes, IPv6 addresses and other configuration parameters to hosts.

Compared with other IPv6 address allocation methods (such as manual configuration and stateless address autoconfiguration), DHCPv6 can:

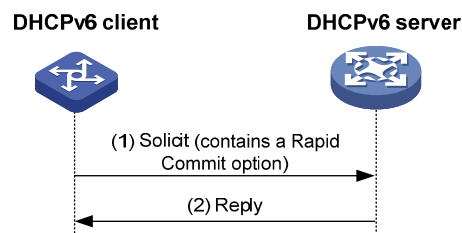
- Record addresses assigned to hosts and assign specific addresses to hosts, thus facilitating network management.
- Assign prefixes to devices, thus facilitating automatic configuration and management of the entire network.
- Assign other configuration parameters, such as the DNS server addresses and domain names, to hosts.

Address/prefix assignment

A process of DHCPv6 address/prefix assignment involves two or four messages. The following describe the detailed processes.

Rapid assignment involving two messages

Figure 61 Rapid assignment involving two messages



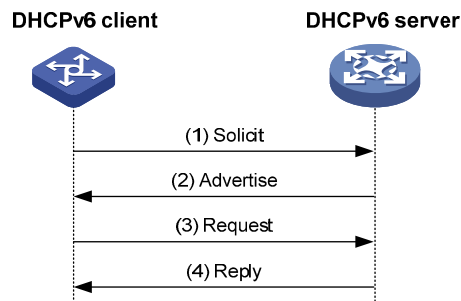
As shown in [Figure 61](#), the rapid assignment involving two messages operates in the following steps.

1. The DHCPv6 client sends out a Solicit message that contains a Rapid Commit option, requesting that rapid assignment of address/prefix and other configuration parameters should be preferred.
2. If the DHCPv6 server supports rapid assignment, it responds with a Reply message containing the assigned IPv6 address/prefix and other configuration parameters. If the DHCPv6 server does not support rapid assignment, [Assignment involving four messages](#) is implemented.

Assignment involving four messages

Figure 62 shows the process of IPv6 address/prefix assignment involving four messages.

Figure 62 Assignment involving four messages



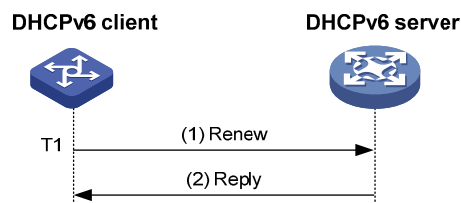
The assignment involving four messages operates in the following steps.

1. The DHCPv6 client sends out a Solicit message, requesting an IPv6 address/prefix and other configuration parameters.
2. If the Solicit message does not contain a Rapid Commit option, or the DHCPv6 server does not support rapid assignment though a Rapid Commit option is contained, the DHCPv6 server responds with an Advertise message, informing the DHCPv6 client of the assignable address/prefix and other configuration parameters.
3. The DHCPv6 client can receive multiple Advertise messages offered by different DHCPv6 servers. It then selects an offer according to the receiving sequence and server priority, and sends a Request message to the selected server for the confirmation of assignment.
4. The DHCPv6 server sends a Reply message to the client, confirming that the address/prefix and other configuration parameters are assigned to the client.

Address/prefix lease renewal

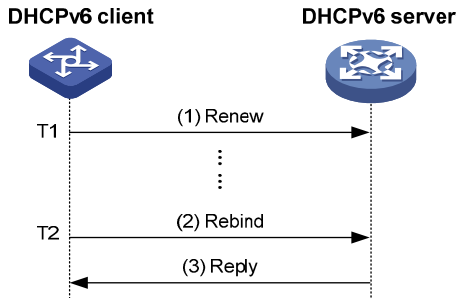
The IPv6 address/prefix assigned by the DHCPv6 server has a lease time, which depends on the valid lifetime. When the valid lifetime of the IPv6 address/prefix expires, the DHCPv6 client cannot use the IPv6 address/prefix any longer. To use the IPv6 address/prefix longer, the DHCPv6 client has to renew the lease time.

Figure 63 Using the Renew message for address/prefix lease renewal



As shown in Figure 63, at T1, the DHCPv6 client unicasts a Renew message to the DHCPv6 server that assigned the IPv6 address/prefix to the DHCPv6 client. HP recommends a value of T1, which is half the preferred lifetime. Then the DHCPv6 server responds with a Reply message, informing that the lease is renewed or not.

Figure 64 Using the Rebind message for address/prefix lease renewal



As shown in [Figure 64](#), if the DHCPv6 client receives no response from the DHCPv6 server after sending out a Renew message at T1, it multicasts a Rebind message to all DHCPv6 servers at T2 (that is, when 80% preferred lifetime expires). Then the DHCPv6 server responds with a Reply message, informing that the lease is renewed or not.

If the DHCPv6 client receives no response from the DHCPv6 servers, the client stops using the address/prefix when the valid lifetime expires.

For more information about the valid lifetime and the preferred lifetime, see [“IPv6 basics configuration.”](#)

Stateless DHCPv6 configuration

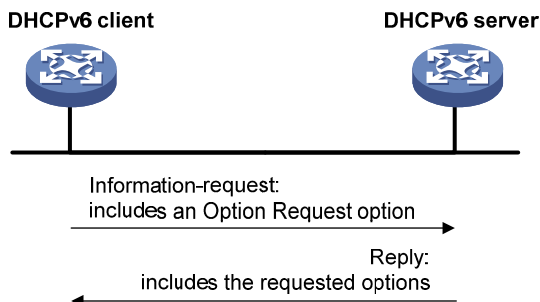
After obtaining an IPv6 address/prefix, a switch can use stateless DHCPv6 to obtain other configuration parameters from a DHCPv6 server. This application is called stateless DHCPv6 configuration.

With an IPv6 address obtained through stateless address autoconfiguration, a switch automatically enables the stateless DHCPv6 function after it receives an RA message with the M flag set to 0 and with the O flag set to 1.

Stateless address autoconfiguration means that a node automatically generates an IPv6 address based on the information obtained through router/prefix discovery. For more information, see [“IPv6 basics configuration.”](#)

Operation

Figure 65 Operation of stateless DHCPv6



As shown in [Figure 65](#), stateless DHCPv6 operates in the following steps.

1. The DHCPv6 client multicasts an Information-request message to the multicast address of all DHCPv6 servers and DHCPv6 relay agents. The Information-request message contains an Option

Request option, specifying the configuration parameters that the client requests from the DHCPv6 server.

2. After receiving the Information-request message, the DHCPv6 server returns the client a Reply message containing the requested configuration parameters.
3. The client checks the Reply message. If the obtained configuration parameters match those requested in the Information-request message, the client performs network configuration with the parameters. If not, the client ignores the configuration parameters. If multiple replies are received, the first received reply is used.

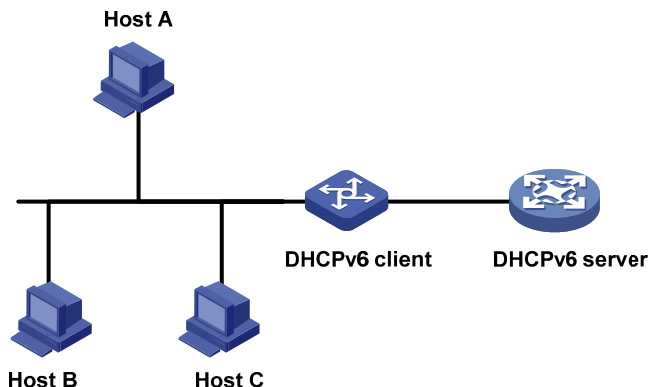
Protocols and standards

- RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*
- RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
- RFC 2462, *IPv6 Stateless Address Autoconfiguration*
- RFC 3633, *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*

DHCPv6 server configuration

Application environment

Figure 66 Typical DHCPv6 server application



As shown in Figure 66, the DHCPv6 server assigns the DHCPv6 client an IPv6 prefix to facilitate IPv6 address management and network configuration. After obtaining the IPv6 prefix, the DHCPv6 client sends an RA message containing the prefix information to the subnet where it resides, so that hosts on the subnet can automatically configure their IPv6 addresses by using the prefix.

A switch serving as a DHCPv6 server assigns DHCPv6 clients IPv6 prefixes, but not IPv6 addresses, and supports DHCPv6 stateless configuration to assign other configuration parameters.

Basic concepts

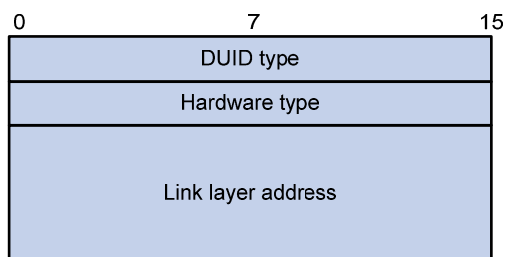
DHCPv6 multicast address

The multicast address FF05::1:3 identifies all DHCPv6 servers on the site-local network. The multicast address FF02::1:2 identifies all DHCPv6 servers and relay agents on the link-local link.

DUID

A DUID uniquely identifies a DHCPv6 switch (DHCPv6 client, server, or relay agent).

Figure 67 Format of DUID-LL



A DUID-LL defined in RFC 3315 is used to identify a DHCPv6 switch. Figure 67 shows the DUID-LL format, where:

- DUID type: The switch supports DUID-LL as the DUID type with the value of 0x0003.
- Hardware type: The switch supports Ethernet as the hardware type with the value of 0x0001.
- Link layer address: Its value is the bridge MAC address of the switch.

IA

Identified by an IAID, an IA provides a construct through which the obtained addresses, prefixes, and other configuration parameters assigned from a server to a client are managed. A client can maintain multiple IAs, each of which is configured on an interface to manage the addresses, prefixes, and other configuration parameters obtained by that interface.

IAID

An IAID uniquely identifies an IA. It is chosen by the client and must be unique among the IAIDs on the client.

PD

The PD is the lease record created by the DHCPv6 server for each assigned prefix. The PD contains information such as the IPv6 prefix, client DUID, IAID, valid lifetime, preferred lifetime, lease expiration time, and the IPv6 address of the requesting client.

Prefix selection process

Upon receiving a request, the DHCPv6 server selects the prefix and other configuration parameters from the address pool that is applied to the interface receiving the request. An address pool can contain the static prefixes configured for specific clients, or have a prefix pool referenced for dynamic assignment from the specific prefix range.

A DHCPv6 server selects a prefix from the address pool according to the following sequence:

1. The desired static prefix with the DUID and IAID matching those of the client
2. The static prefix with the DUID and IAID matching those of the client
3. The desired static prefix with the DUID matching the client's DUID and with no client IAID specified
4. The static prefix with the DUID matching the client's DUID and with no client IAID specified
5. The desired idle prefix in the prefix pool
6. An idle prefix in the prefix pool

Configuration task list

| Task | Remarks |
|---|-----------|
| Enabling the DHCPv6 server | Required. |
| Creating a prefix pool | Required. |
| Configuring a DHCPv6 address pool | Required. |
| Applying the address pool to an interface | Required. |

Configuration prerequisites

Before you configure the DHCPv6 server, enable IPv6 by using **ipv6**. For more information about **ipv6**, see the chapter “IPv6 basics configuration commands.”

Enabling the DHCPv6 server

| Step | Command | Remarks |
|---------------------------------------|--------------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable the DHCPv6 server function. | ipv6 dhcp server enable | Required. Disabled by default. |

Creating a prefix pool

| Step | Command | Remarks |
|--------------------------|--|---|
| 1. Enter system view. | system-view | — |
| 2. Create a prefix pool. | ipv6 dhcp prefix-pool prefix-pool-number prefix prefix/prefix-len assign-len assign-len | Required. Not configured by default. |

Configuring a DHCPv6 address pool

Configure the prefix and other configuration parameters (such as the DNS server address, domain name, SIP server address, and domain name of the SIP server) in a DHCPv6 address pool, for the DHCPv6 server to assign them to DHCPv6 clients.

To configure a DHCPv6 address pool:

| Step | Command | Remarks |
|---|---|---|
| 1. Enter system view. | system-view | — |
| 2. Create a DHCPv6 address pool and enter DHCPv6 address pool view. | ipv6 dhcp pool pool-number | Required. Not configured by default. |
| 3. Configure a static prefix. | static-bind prefix prefix/ prefix-len duid duid [iaid iaaid] [preferred-lifetime preferred-lifetime valid-lifetime valid-lifetime] | Required. Use either command. |
| 4. Apply a prefix pool to the address pool. | prefix-pool prefix-pool-number [preferred-lifetime preferred-lifetime valid-lifetime valid-lifetime] | No prefix is specified by default. |
| 5. Configure a DNS server address. | dns-server ipv6-address | Optional. Not configured by default. |

| Step | Command | Remarks |
|---|--|---|
| 6. Configure a domain name. | domain-name <i>domain-name</i> | Optional. Not configured by default. |
| 7. Configure the IPv6 address or domain name of a SIP server. | sip-server { address <i>ipv6-address</i> domain-name <i>domain-name</i> } | Optional. Not configured by default. |

Only one prefix pool can be referenced by an address pool.

A non-existing prefix pool can be referenced by an address pool. However, no prefix is available in the prefix pool for dynamic prefix assignment until the prefix pool is created.

You cannot modify the prefix pool referenced by an address pool, or the preferred lifetime or valid lifetime by using **prefix-pool**. You need to remove the configuration before have another prefix pool referenced by the address pool, or modify the preferred lifetime and valid lifetime.

Configure up to eight DNS server addresses, one domain name, eight SIP server addresses, and eight SIP server domain names in an address pool.

Applying the address pool to an interface

After an address pool is applied to an interface, a prefix and other configuration parameters can be selected from the address pool and assigned to the DHCPv6 client requesting through the interface.

To apply an address pool to an interface:

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface <i>interface-type</i> <i>interface-number</i> | — |
| 3. Apply the DHCPv6 address pool to the interface. | ipv6 dhcp server apply pool <i>pool-number</i> [allow-hint preference <i>preference-value</i> rapid-commit] * | Required. Not configured by default. |

An interface cannot serve as a DHCPv6 server and DHCPv6 relay agent at the same time.

HP does not recommend that you enable DHCPv6 server and DHCPv6 client on the same interface.

Only one address pool can be applied to an interface.

A non-existing address pool can be applied to an interface. However, the server cannot assign any prefix or other configuration information from the address pool until the address pool is created.

You cannot modify the address pool applied to an interface or parameters such as the server priority by using **ipv6 dhcp server apply pool**. You need to remove the applied address pool before apply another address pool to the interface or modify parameters such as the server priority.

Displaying and maintaining the DHCPv6 server

| Task | Command | Remarks |
|--|--|------------------------|
| Display the DUID of the local device. | display ipv6 dhcp duid [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display the DHCPv6 address pool information. | display ipv6 dhcp pool [<i>pool-number</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display the prefix pool information. | display ipv6 dhcp prefix-pool [<i>prefix-pool-number</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display the DHCPv6 server configuration information. | display ipv6 dhcp server [<i>interface interface-type interface-number</i>] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display the PD information. | display ipv6 dhcp server pd-in-use { all pool <i>pool-number</i> prefix <i>prefix/prefix-len</i> prefix-pool <i>prefix-pool-number</i> } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display packet statistics on the DHCPv6 server. | display ipv6 dhcp server statistics [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Clear PD information on the DHCPv6 server. | reset ipv6 dhcp server pd-in-use { all pool <i>pool-number</i> prefix <i>prefix/prefix-len</i> } | Available in user view |
| Clear packets statistics on the DHCPv6 server. | reset ipv6 dhcp server statistics | Available in user view |

DHCPv6 server configuration example

Network requirements

As shown in [Figure 68](#), the switch serves as a DHCPv6 server, and assigns the IPv6 prefix, DNS server address, domain name, SIP server address, and the domain name of the SIP server to the DHCPv6 clients. The IPv6 address of the switch is 1::1/64.

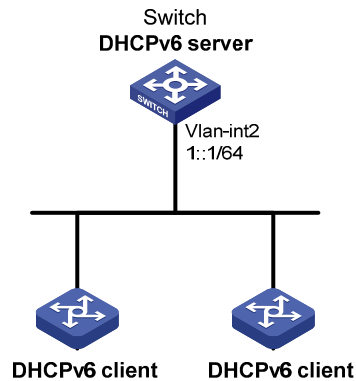
The switch assigns prefix 2001:0410:0201::/48 to the clients whose DUID is 00030001CA0006A40000, and assigns prefixes ranging from 2001:0410::/48 to 2001:0410:FFFF::/48 (excluding 2001:0410:0201::/48) to other clients. The DNS server address is 2::2:3. The DHCPv6 clients reside in domain aaa.com. The SIP server address is 2:2::4, and the domain name of the SIP server is bbb.com.

Configuration considerations

To configure the DHCPv6 server.

- Enable IPv6 and DHCPv6 server.
- Create a prefix pool containing prefix 2001:0410::/32 with the length of the assigned prefix being 48, so that the server assigns clients the prefixes ranging 2001:0410::/48 to 2001:0410:FFFF::/48.
- Create an address pool. Configure a static prefix in the address pool and have the prefix pool referenced by the address pool. Configure other configuration parameters.
- Apply the address pool to the interface through which the server is connected to the clients.

Figure 68 DHCPv6 server configuration



Configuration procedure

Enable IPv6 and DHCPv6 server.

```
<Switch> system-view
[Switch] ipv6
[Switch] ipv6 dhcp server enable
```

Configure the IPv6 address of VLAN-interface 2.

```
[Switch] interface vlan-interface 2
[Switch-Vlan-interface2] ipv6 address 1::1/64
[Switch-Vlan-interface2] quit
```

Create and configure prefix pool 1.

```
[Switch] ipv6 dhcp prefix-pool 1 prefix 2001:0410::/32 assign-len 48
```

Create address pool 1.

```
[Switch] ipv6 dhcp pool 1
```

Apply prefix pool 1 to address pool 1, and set the preferred lifetime to one day, the valid lifetime to three days.

```
[Switch-ipv6-dhcp-pool-1] prefix-pool 1 preferred-lifetime 86400 valid-lifetime 259200
```

Configure static prefix 2001:0410:0201::/48 in address pool 1, and set the client DUID as 00030001CA0006A40000, the preferred lifetime to one day, and the valid lifetime to three days.

```
[Switch-ipv6-dhcp-pool-1] static-bind prefix 2001:0410:0201::/48 duid
00030001CA0006A40000 preferred-lifetime 86400 valid-lifetime 259200
```

Configure the DNS server address as 2:2::3.

```
[Switch-ipv6-dhcp-pool-1] dns-server 2:2::3
```

Configure the domain name as aaa.com.

```
[Switch-ipv6-dhcp-pool-1] domain-name aaa.com
```

Configure the SIP server address as 2:2::4, and the domain name of the SIP server as bbb.com.

```
[Switch-ipv6-dhcp-pool-1] sip-server address 2:2::4
[Switch-ipv6-dhcp-pool-1] sip-server domain-name bbb.com
[Switch-ipv6-dhcp-pool-1] quit
```

Apply address pool 1 to VLAN-interface 2, configure the address pool to support the desired prefix assignment and rapid prefix assignment, and set the precedence to the highest.

```
[Switch] interface vlan-interface 2
```

```
[Switch-Vlan-interface2] ipv6 dhcp server apply pool 1 allow-hint preference 255 rapid-commit
```

Verification

After the preceding configuration is complete, display the DHCPv6 server configuration information on VLAN-interface 2.

```
[Switch-Vlan-interface2] display ipv6 dhcp server interface vlan-interface 2
Using pool: 1
Preference value: 255
Allow-hint: Enabled
Rapid-commit: Enabled
```

Display the information of address pool 1.

```
[Switch-Vlan-interface2] display ipv6 dhcp pool 1
DHCPv6 pool: 1
  Static bindings:
    DUID: 00030001CA0006A40000
    IAID: A1A1A1A1
    Prefix: 2001:410:201::/48
      preferred lifetime 86400, valid lifetime 2592000
    Prefix pool: 1
      preferred lifetime 86400, valid lifetime 2592000
    DNS server address:
      2:2::3
    Domain name: aaa.com
    SIP server address:
      2:2::4
    SIP server domain name:
      bbb.com
```

Display the information of prefix pool 1.

```
[Switch-Vlan-interface2] display ipv6 dhcp prefix-pool 1
Prefix: 2001:410::/32
Assigned length: 48
Total prefix number: 65536
Available: 65535
In-use: 0
Static: 1
```

After the client whose DUID is 00030001CA0006A40000 obtains an IPv6 prefix, display the PD information on the DHCPv6 server.

```
[Switch-Vlan-interface2] display ipv6 dhcp server pd-in-use all
Total number = 1
Prefix                                Type      Pool Lease-expiration
2001:410:201::/48                     Static(C) 1      Jul 10 2009 19:45:01
```


After the other client obtains an IPv6 prefix, display the PD information on the DHCPv6 server.

```
[Switch-Vlan-interface2] display ipv6 dhcp server pd-in-use all
```

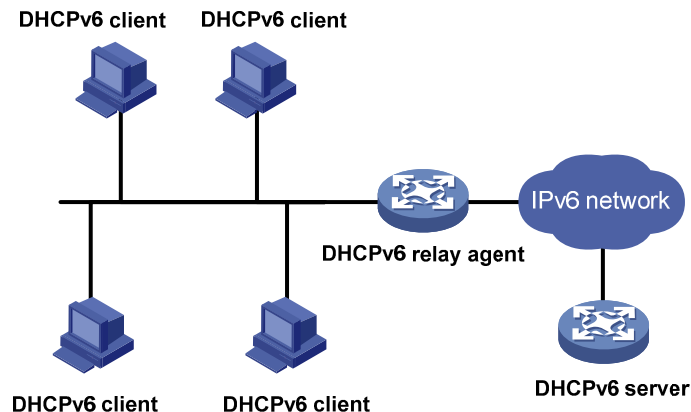
```
Total number = 2
```

| Prefix | Type | Pool | Lease-expiration |
|-------------------|-----------|------|----------------------|
| 2001:410:201::/48 | Static(C) | 1 | Jul 10 2009 19:45:01 |
| 2001:410::/48 | Auto(C) | 1 | Jul 10 2009 20:44:05 |

DHCPv6 relay agent configuration

Application environment

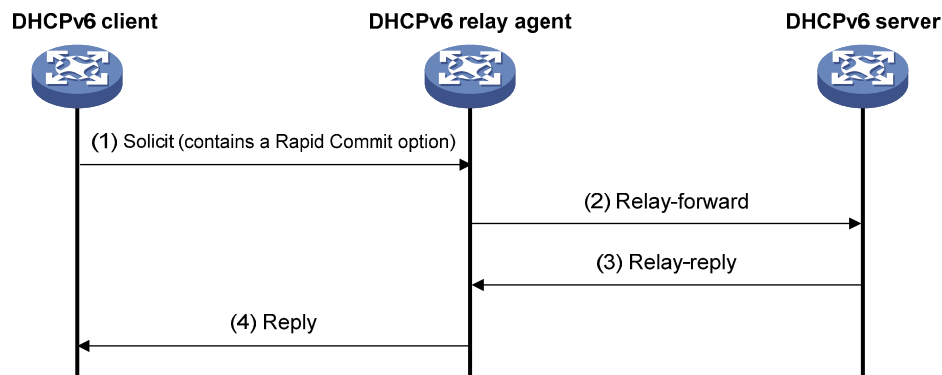
Figure 69 Typical DHCPv6 relay agent application



A DHCPv6 client usually uses a multicast address to contact the DHCPv6 server on the local link to obtain an IPv6 address and other configuration parameters. As shown in Figure 69, if the DHCPv6 server resides on another subnet, the DHCPv6 client can contact the server via a DHCPv6 relay agent. Thus, you do not need to deploy a DHCPv6 server on each subnet.

Operation

Figure 70 Operating process of a DHCPv6 relay agent



Take the process of rapid assignment involving two messages as an example. Figure 70 shows how the DHCPv6 client obtains the IPv6 address and other network configuration parameters from the DHCPv6 server through the DHCPv6 relay agent.

1. The DHCPv6 client sends a Solicit message containing the Rapid Commit option to the multicast address FF02::1:2 of all DHCPv6 servers and relay agents.
2. After receiving the Solicit message, the DHCPv6 relay agent encapsulates the message into the Relay Message option of a Relay-forward message, and sends the message to the DHCPv6 server.

3. After obtaining the Solicit message from the Relay-forward message, the DHCPv6 server selects an IPv6 address and other required parameters, and adds them to the reply which is encapsulated within the Relay Message option of a Relay-reply message. The DHCPv6 server then sends the Relay-reply message to the DHCPv6 relay agent.
4. The DHCPv6 relay agent obtains the reply from the Relay-reply message and sends the reply to the DHCPv6 client.

Then the DHCPv6 client uses the IPv6 address and other network parameters assigned by the DHCPv6 server to perform network configuration.

Configuring the DHCPv6 relay agent

Upon receiving a Solicit message from a DHCPv6 client, the interface that operates as a DHCPv6 relay agent encapsulates the request into a Relay-forward message and forwards the message to the specified DHCPv6 server, which then assigns an IPv6 address and other configuration parameters to the DHCPv6 client.

Configuration prerequisites

Before you configure the DHCPv6 relay agent, enable IPv6 by using **ipv6** in system view.

Configuration procedure

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Enable DHCPv6 relay agent on the interface and specify a DHCPv6 server. | ipv6 dhcp relay server-address ipv6-address [interface interface-type interface-number] | Required. By default, DHCPv6 relay agent is disabled and no DHCPv6 server is specified on the interface. |

Executing **ipv6 dhcp relay server-address** repeatedly can specify multiple DHCPv6 servers, and up to eight DHCPv6 servers can be specified for an interface. After receiving requests from DHCPv6 clients, the DHCPv6 relay agent forwards the requests to all specified DHCPv6 servers.

If the DHCPv6 server address is a link-local address or link-scoped multicast address on the local link, you need to specify an outgoing interface using the **interface** keyword in **ipv6 dhcp relay server-address**; otherwise, DHCPv6 packets can fail to be forwarded to the DHCPv6 server.

After you remove all specified DHCPv6 servers from an interface with **undo ipv6 dhcp relay server-address**, DHCPv6 relay agent is disabled on the interface.

An interface cannot serve as a DHCPv6 relay agent and DHCPv6 server at the same time.

HP recommends you not enable the DHCPv6 relay agent and DHCPv6 client on the same interface.

Displaying and maintaining the DHCPv6 relay agent

| Task | Command | Remarks |
|--|---|------------------------|
| Display the DUID of the local device. | display ipv6 dhcp duid [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display DHCPv6 server addresses specified on the DHCPv6 relay agent. | display ipv6 dhcp relay server-address { all interface <i>interface-type interface-number</i> } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display packet statistics on the DHCPv6 relay agent. | display ipv6 dhcp relay statistics [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Clear packets statistics on the DHCPv6 relay agent. | reset ipv6 dhcp relay statistics | Available in user view |

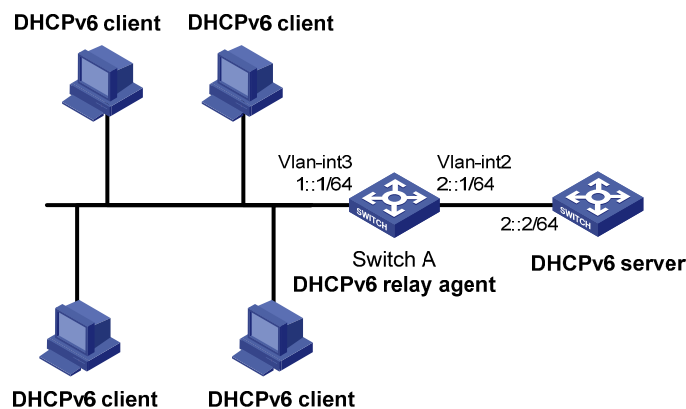
DHCPv6 relay agent configuration example

Network requirements

As shown in Figure 71, the network address prefix of DHCPv6 clients is 1::/64, and the IPv6 address of the DHCPv6 server is 2::2/64. The DHCPv6 client and server need to communicate via a DHCPv6 relay agent (Switch A).

Switch A acts as the gateway of network 1::/64. It sends RA messages to notify the hosts to obtain IPv6 addresses and other configuration parameters through DHCPv6.

Figure 71 DHCPv6 relay agent configuration



Configuration procedure

1. Configure Switch A as a DHCPv6 relay agent

Enable the IPv6 packet forwarding function.

```
<SwitchA> system-view
[SwitchA] ipv6
```

Configure the IPv6 addresses of VLAN-interface 2 and VLAN-interface 3 respectively.

```
[SwitchA] interface vlan-interface 2
[SwitchA-Vlan-interface2] ipv6 address 2::1 64
[SwitchA-Vlan-interface2] quit
[SwitchA] interface vlan-interface 3
[SwitchA-Vlan-interface3] ipv6 address 1::1 64
```

Enable DHCPv6 relay agent and specify the DHCPv6 server address on VLAN-interface 3.

```
[SwitchA-Vlan-interface3] ipv6 dhcp relay server-address 2::2
```

2. Configure Switch A as a gateway

Enable Switch A to send RA messages and turn on the M and O flags.

```
[SwitchA-Vlan-interface3] undo ipv6 nd ra halt
[SwitchA-Vlan-interface3] ipv6 nd autoconfig managed-address-flag
[SwitchA-Vlan-interface3] ipv6 nd autoconfig other-flag
```

Verification

After completing the configurations, display DHCPv6 server address information on Switch A.

```
[SwitchA-Vlan-interface3] display ipv6 dhcp relay server-address all
Interface: Vlan3
Server address(es)                               Output Interface
2::2
```

Display packet statistics on the DHCPv6 relay agent.

```
[SwitchA-Vlan-interface3] display ipv6 dhcp relay statistics
Packets dropped          : 0
  Error                  : 0
  Excess of rate limit   : 0
Packets received        : 14
  SOLICIT                : 0
  REQUEST                : 0
  CONFIRM                : 0
  RENEW                  : 0
  REBIND                 : 0
  RELEASE                : 0
  DECLINE                : 0
  INFORMATION-REQUEST   : 7
  RELAY-FORWARD          : 0
  RELAY-REPLY            : 7
Packets sent            : 14
  ADVERTISE              : 0
  RECONFIGURE            : 0
  REPLY                  : 7
  RELAY-FORWARD          : 7
  RELAY-REPLY            : 0
```

DHCPv6 client configuration

Serving as a DHCPv6 client, the switch only supports stateless DHCPv6 configuration and can only obtain other network configuration parameters, except the IPv6 address and prefix from the DHCPv6 server.

With an IPv6 address obtained through stateless address autoconfiguration, the switch automatically enables the stateless DHCPv6 function after it receives an RA message with the M flag set to 0 and the O flag set to 1.

Configuration prerequisites

To make the DHCPv6 client obtain configuration parameters through stateless DHCPv6 configuration successfully, make sure that the DHCPv6 server is available.

Configuration procedure

| Step | Command | Remarks |
|---|--|-----------|
| 1. Enter system view. | system-view | — |
| 2. Enable the IPv6 packet forwarding function. | ipv6 | Required. |
| 3. Enter interface view. | interface interface-type interface-number | — |
| 4. Enable IPv6 stateless address autoconfiguration. | ipv6 address auto | Required. |

For more information about **ipv6 address auto**, see “[IPv6 basics configuration](#).”

HP recommends you not enable the DHCPv6 client and DHCPv6 server, or the DHCPv6 client and DHCPv6 relay agent on the same interface at the same time.

Displaying and maintaining the DHCPv6 client

| Task | Command | Remarks |
|---------------------------------------|---|-----------------------|
| Display DHCPv6 client information. | display ipv6 dhcp client [interface interface-type interface-number] [{ begin exclude include } regular-expression] | Available in any view |
| Display DHCPv6 client statistics. | display ipv6 dhcp client statistics [interface interface-type interface-number] [{ begin exclude include } regular-expression] | Available in any view |
| Display the DUID of the local device. | display ipv6 dhcp duid [{ begin exclude include } regular-expression] | Available in any view |

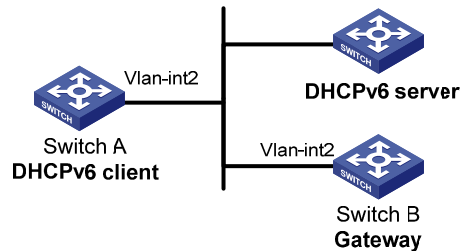
Stateless DHCPv6 configuration example

Network requirements

As shown in [Figure 72](#), through stateless DHCPv6, Switch A obtains the DNS server address, domain name, and other information from the server.

Switch B acts as the gateway to send RA messages periodically.

Figure 72 Stateless DHCPv6 configuration



Configuration procedure

1. Configure Switch B

Enable the IPv6 packet forwarding function.

```
<SwitchB> system-view  
[SwitchB] ipv6
```

Configure the IPv6 address of VLAN-interface 2.

```
[SwitchB] interface vlan-interface 2  
[SwitchB-Vlan-interface2] ipv6 address 1::1 64
```

Set the O flag in the RA messages to 1.

```
[SwitchB-Vlan-interface2] ipv6 nd autoconfig other-flag
```

Enable Switch B to send RA messages.

```
[SwitchB-Vlan-interface2] undo ipv6 nd ra halt
```

2. Configure Switch A

Enable the IPv6 packet forwarding function.

```
<SwitchA> system-view  
[SwitchA] ipv6
```

Enable stateless IPv6 address autoconfiguration on VLAN-interface 2.

```
[SwitchA] interface vlan-interface 2  
[SwitchA-Vlan-interface2] ipv6 address auto
```

With this command executed, if VLAN-interface 2 has no IPv6 address configured, Switch A automatically generates a link-local address, and send an RS message, requesting the gateway (Switch B) to reply with an RA message immediately.

Verification

After receiving an RA message with the M flag set to 0 and the O flag set to 1, Switch A automatically enables the stateless DHCPv6 function.

Use **display ipv6 dhcp client** to view the current client configuration information. If the client successfully obtains configuration information from the server, the following information is displayed.

```
[SwitchA-Vlan-interface2] display ipv6 dhcp client interface vlan-interface 2
Vlan-interface2 is in stateless DHCPv6 client mode
State is OPEN
Preferred Server:
  Reachable via address      : FE80::213:7FFF:FEF6:C818
  DUID                      : 0003000100137ff6c818
  DNS servers                : 1:2:3::5
                             1:2:4::7
  Domain names              : abc.com
                             Sysname.com
```

Use **display ipv6 dhcp client statistics** to view the current client statistics.

```
[SwitchA-Vlan-interface2] display ipv6 dhcp client statistics
Interface                : Vlan-interface2
Packets Received         : 1
  Reply                  : 1
  Advertise              : 0
  Reconfigure            : 0
  Invalid                : 0
Packets Sent             : 5
  Solicit                : 0
  Request                : 0
  Confirm                : 0
  Renew                  : 0
  Rebind                 : 0
  Information-request     : 5
  Release                 : 0
  Decline                 : 0
```


DHCPv6 snooping configuration

A DHCPv6 snooping device does not work if it is between a DHCPv6 relay agent and a DHCPv6 server. The DHCPv6 snooping device works when it is between a DHCPv6 client and a DHCPv6 relay agent or between a DHCPv6 client and a DHCPv6 server.

Configure only Layer 2 Ethernet interfaces or Layer 2 aggregate interfaces as DHCPv6 snooping trusted ports. For more information about aggregate interfaces, see *Layer 2—LAN Switching Configuration Guide*.

Overview

As a DHCPv6 security feature, DHCPv6 snooping can implement the following functions.

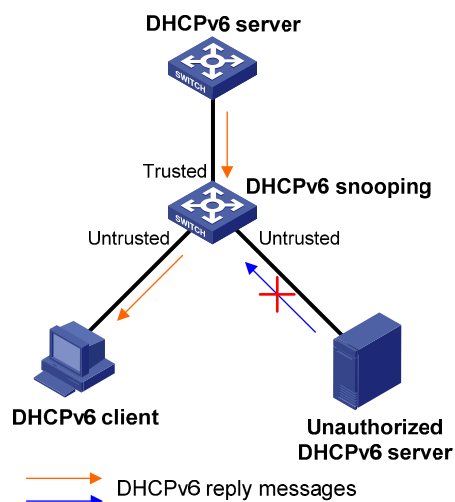
- Ensuring DHCPv6 clients to obtain IPv6 addresses from authorized DHCPv6 servers
- Recording IP-to-MAC mappings of DHCPv6 clients

Ensuring DHCPv6 clients to obtain IPv6 addresses from authorized DHCPv6 servers

If there is an unauthorized DHCPv6 server on a network, DHCPv6 clients can obtain invalid IPv6 addresses and network configuration parameters, and cannot communicate with other network devices. With DHCPv6 snooping, the ports of a device can be configured as trusted or untrusted, ensuring the clients to obtain IPv6 addresses from authorized DHCPv6 servers.

- Trusted: A trusted port forwards DHCPv6 messages normally.
- Untrusted: An untrusted port discards the reply messages from any DHCPv6 server.

Figure 73 Trusted and untrusted ports



A DHCPv6 snooping device's port that is connected to an authorized DHCPv6 server, DHCPv6 relay agent, or another DHCPv6 snooping device should be configured as a trusted port to forward reply messages from the authorized DHCPv6 server, whereas other ports are configured as untrusted so that the

DHCPv6 client can obtain an IPv6 address from the authorized DHCPv6 server only. As shown in [Figure 73](#), configure the port that connects to the DHCPv6 server as a trusted port, and other ports as untrusted.

Recording IP-to-MAC mappings of DHCPv6 clients

DHCPv6 snooping reads DHCPv6 messages to create and update DHCPv6 snooping entries, including MAC addresses of clients, IPv6 addresses obtained by the clients, ports that connect to DHCPv6 clients, and VLANs to which the ports belong. use **display ipv6 dhcp snooping user-binding** to view the IPv6 address obtained by each client and manage and monitor the clients' IPv6 addresses.

Enabling DHCPv6 snooping

To allow clients to obtain IPv6 addresses from an authorized DHCPv6 server, enable DHCPv6 snooping globally and configure trusted and untrusted ports properly. To record DHCPv6 snooping entries for a VLAN, enable DHCPv6 snooping for the VLAN.

To enable DHCPv6 snooping:

| Step | Command | Remarks |
|---|---------------------------------------|-----------------------------------|
| 1. Enter system view. | system-view | — |
| 2. Enable DHCPv6 snooping globally. | ipv6 dhcp snooping enable | Required. Disabled by default. |
| 3. Enter VLAN view. | vlan vlan-id | — |
| 4. Enable DHCPv6 snooping for the VLAN. | ipv6 dhcp snooping vlan enable | Optional. Disabled by default. |

Configuring a DHCPv6 snooping trusted port

After enabling DHCPv6 snooping globally, specify trusted and untrusted ports for a VLAN as needed. A DHCPv6 snooping trusted port normally forwards DHCPv6 packets it receives. A DHCPv6 snooping untrusted port discards any DHCPv6 reply message received from a DHCPv6 server. Upon receiving a DHCPv6 request from a client in the VLAN, the DHCPv6 snooping device forwards the packet through trusted ports rather than any untrusted port in the VLAN, thus reducing network traffic.

To configure a DHCPv6 snooping trusted port:

| Step | Command | Remarks |
|-----------------------------------|--|---|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure the port as trusted. | ipv6 dhcp snooping trust | Required. By default, all ports of the device with DHCPv6 snooping globally enabled are untrusted. |

You need to specify a port connected to an authorized DHCPv6 server as trusted to make sure that DHCPv6 clients can obtain valid IPv6 addresses. The trusted port and the ports connected to the DHCPv6 clients must be in the same VLAN.

If a Layer 2 Ethernet interface is added to an aggregation group, the DHCPv6 snooping configuration of the interface does not take effect until the interface quits the aggregation group.

Configuring the maximum number of DHCPv6 Snooping entries an interface can learn

| Step | Command | Remarks |
|--|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enter interface view. | interface interface-type interface-number | — |
| 3. Configure the maximum number of DHCPv6 snooping entries that the interface can learn. | ipv6 dhcp snooping max-learning-num number | Optional. By default, the number of DHCPv6 snooping entries learned by an interface is not limited. |

Displaying and maintaining DHCPv6 snooping

| Task | Command | Remarks |
|--|--|------------------------|
| Display DHCPv6 snooping trusted ports. | display ipv6 dhcp snooping trust [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display DHCPv6 snooping entries. | display ipv6 dhcp snooping user-binding { <i>ipv6-address</i> dynamic } [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Clear DHCPv6 snooping entries. | reset ipv6 dhcp snooping user-binding { <i>ipv6-address</i> dynamic } | Available in user view |

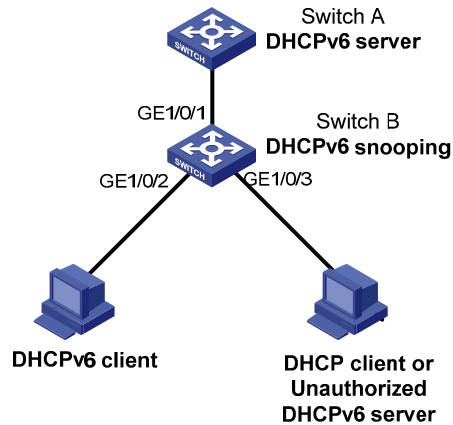
DHCPv6 snooping configuration example

Network requirements

As shown in [Figure 74](#), Switch B is connected to a DHCPv6 server (Switch A) through GigabitEthernet 1/0/1, and is connected to DHCPv6 clients through GigabitEthernet1/0/2 and GigabitEthernet1/0/3. These three interfaces belong to VLAN 2. Configure Switch B to do the following:

- Forwarding DHCPv6 reply messages only received on GigabitEthernet1/0/1.
- Recording the IP-to-MAC mappings for DHCPv6 clients.

Figure 74 Network diagram for DHCPv6 snooping configuration



Configuration procedure

Enable DHCPv6 snooping globally.

```
<SwitchB> system-view
[SwitchB] ipv6 dhcp snooping enable
```

Add GigabitEthernet1/0/1, GigabitEthernet1/0/2, and GigabitEthernet1/0/3 to VLAN 2.

```
[SwitchB] vlan 2
[SwitchB-vlan2] port GigabitEthernet 1/0/1 GigabitEthernet 1/0/2 GigabitEthernet 1/0/3
```

Enable DHCPv6 snooping for VLAN 2.

```
[SwitchB-vlan2] ipv6 dhcp snooping vlan enable
[SwitchB] quit
```

Configure GigabitEthernet1/0/1 as a DHCPv6 snooping trusted port.

```
[SwitchB] interface GigabitEthernet 1/0/1
[SwitchB-GigabitEthernet1/0/1] ipv6 dhcp snooping trust
```

Verify the configuration.

After completing the configuration, connect GigabitEthernet1/0/2 to a DHCPv6 client, GigabitEthernet1/0/1 to a DHCPv6 server (Switch A), and GigabitEthernet1/0/3 to an unauthorized DHCPv6 server. The DHCPv6 client obtains an IPv6 address from Switch A, but cannot obtain any IPv6 address from the unauthorized DHCPv6 server. use **display ipv6 dhcp snooping user-binding** to view the DHCPv6 snooping entries on Switch B.

Tunneling configuration

Tunneling is an encapsulation technology, which uses one network protocol to encapsulate packets of another network protocol and transfer them over the network. A tunnel is a virtual point-to-point connection providing a channel to transfer encapsulated packets. Packets are encapsulated and de-encapsulated at both ends of a tunnel. Tunneling refers to the whole process from data encapsulation to data transfer to data de-encapsulation.

Tunneling provides the following features:

- Transition techniques, such as IPv6 over IPv4 tunneling, to interconnect IPv4 and IPv6 networks.
- VPNs for guaranteeing communication security, such as GRE, DVPN, and IPsec tunneling.
- Traffic engineering, such as MPLS TE to prevent network congestion.

The preceding tunneling technologies require that you create virtual Layer 3 interfaces (tunnel interfaces) at both ends of a tunnel, so that devices at both ends can send, identify, and process packets transferred through the tunnel.

Unless otherwise specified, the term *tunnel* used throughout this document refers to an IPv4/IPv6 transition tunnel, IPv4 over IPv4 tunnel or IPv6 over IPv6 tunnel.

IPv4/IPv6 tunnels

Internet expansion results in scarce IPv4 addresses. The technologies such as temporary IPv4 address allocation and NAT relieve the problem of IPv4 address shortage to some extent. However, these technologies not only increase the overhead in address resolution and processing, but can also lead to upper-layer application failures. Furthermore, even with these technologies, IPv4 addresses are eventually used up.

IPv6 adopting the 128-bit addressing scheme completely solves the above problem. Since significant improvements have been made in address space, security, network management, mobility, and QoS, IPv6 becomes one of the core standards for the next generation Internet protocol. IPv6 is compatible with all protocols except IPv4 in the TCP/IP suite. Therefore, IPv6 can completely take the place of IPv4.

Before IPv6 becomes the dominant protocol, networks using the IPv6 protocol stack are expected to communicate with the Internet using IPv4. Therefore, an IPv6-IPv4 interworking technology must be developed to ensure the smooth transition from IPv4 to IPv6. In addition, the interworking technology should provide efficient, seamless information transfer. Multiple transition technologies and interworking solutions are available. With their own characteristics, they are used to solve communication problems in different transition stages under different environments.

There are three major transition technologies, dual stack (RFC 2893), tunneling (RFC 2893), and NAT-PT (RFC 2766).

For more information about dual stack, see [“IPv6 basics configuration.”](#)

The switch does not support NAT-PT.

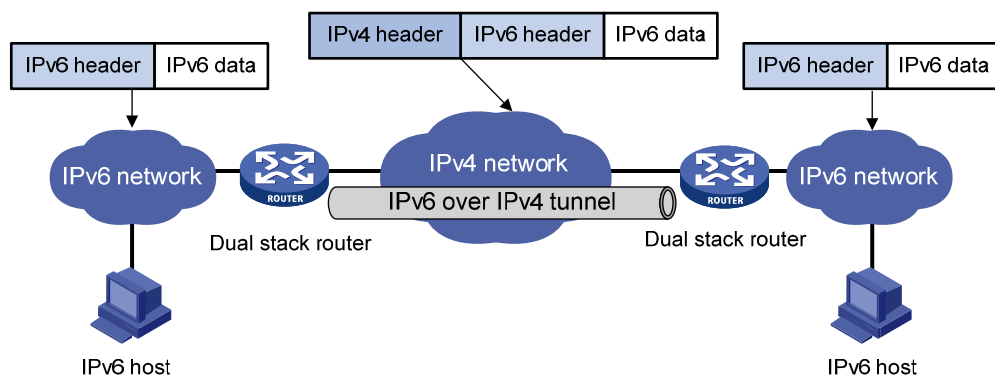
IPv6 over IPv4 tunnel

Implementation

The IPv6 over IPv4 tunneling mechanism adds an IPv4 header to IPv6 data packets so that IPv6 packets can pass an IPv4 network through a tunnel to realize interworking between isolated IPv6 networks, as shown in Figure 75.

The devices at both ends of an IPv6 over IPv4 tunnel must support the IPv4/IPv6 dual stack.

Figure 75 IPv6 over IPv4 tunnel



The IPv6 over IPv4 tunnel processes packets in the following ways.

1. A host in the IPv6 network sends an IPv6 packet to the device at the source end of the tunnel.
2. After determining according to the routing table that the packet must be forwarded through the tunnel, the device at the source end of the tunnel encapsulates the IPv6 packet with an IPv4 header and forwards it through the physical interface of the tunnel.
3. The encapsulated packet goes through the tunnel to reach the device at the destination end of the tunnel. The device at the destination end de-encapsulates the packet if the destination address of the encapsulated packet is the device itself.
4. The destination device forwards the packet according to the destination address in the de-encapsulated IPv6 packet. If the destination address is the device itself, the device forwards the IPv6 packet to the upper-layer protocol for processing.

Configured tunnel and automatic tunnel

An IPv6 over IPv4 tunnel can be established between two hosts, a host and a device, and two devices, and between devices. The tunnel destination must forward packets if the tunnel destination is not the final destination of the IPv6 packet.

Tunnels are divided into configured tunnels and automatic tunnels depending on how the IPv4 address of the tunnel destination is acquired.

- If the destination address of an IPv6 over IPv4 tunnel cannot be acquired from the destination address of IPv6 packets, it must be configured manually. Such a tunnel is called a configured tunnel.
- If the interface address of an IPv6 over IPv4 tunnel has an IPv4 address embedded into an IPv6 address, the IPv4 address of the tunnel destination can be acquired automatically. Such a tunnel is called an automatic tunnel.

Type

According to the way an IPv6 packet is encapsulated, IPv6 over IPv4 tunnels are divided into the following modes, as shown in Table 11.

Table 11 IPv6 over IPv4 tunnel modes and key parameters

| Tunnel type | Tunnel mode | Tunnel source/destination address | Tunnel interface address type |
|----------------------------|-----------------------|---|---|
| Manually configured tunnel | IPv6 manual tunneling | The source/destination IP address is a manually configured IPv4 address. | IPv6 address |
| | 6to4 tunneling | The source IP address is a manually configured IPv4 address. The destination IP address need not be configured. | 6to4 address, in the format of 2002:IPv4-source-address::/48 |
| Automatic tunnel | ISATAP tunneling | The source IP address is a manually configured IPv4 address. The destination IP address need not be configured. | ISATAP address, in the format of Prefix:0:5EFE:IPv4-source-address/64 |

1. IPv6 manually configured tunnel

A manually configured tunnel is a point-to-point link. Each link is a separate tunnel. IPv6 manually configured tunnels are mainly used to provide stable connections for regular secure communication between border routers or between border routers and hosts for access to remote IPv6 networks.

2. 6to4 tunnel

An automatic 6to4 tunnel is a point-to-multipoint tunnel and is used to connect multiple isolated IPv6 networks over an IPv4 network to remote IPv6 networks. The embedded IPv4 address in an IPv6 address is used to automatically acquire the destination IPv4 address of the tunnel.

The automatic 6to4 tunnel adopts 6to4 addresses. The address format is 2002:abcd:efgh:subnet number::interface ID/64, where 2002 represents the fixed IPv6 address prefix, and abcd:efgh represents the 32-bit globally unique source IPv4 address of the 6to4 tunnel, in hexadecimal notation. For example, 1.1.1.1 can be represented by 0101:0101. The part that follows 2002:abcd:efgh uniquely identifies a host in a 6to4 network. The tunnel destination is automatically determined by the embedded IPv4 address, which makes it easy to create a 6to4 tunnel.

Because the 16-bit subnet number of the 64-bit address prefix in 6to4 addresses can be customized and the first 48 bits in the address prefix are fixed to a permanent value and the IPv4 address of the tunnel source or destination, it is possible that IPv6 packets can be forwarded by the tunnel.

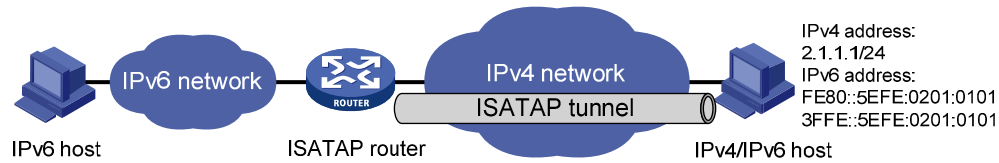
3. ISATAP tunnel

With the application of the IPv6 technology, there will be more and more IPv6 hosts in the existing IPv4 network. The ISATAP tunneling technology provides a satisfactory solution for IPv6 application. An ISATAP tunnel is a point-to-multipoint automatic tunnel. The destination of a tunnel can automatically be acquired from the embedded IPv4 address in the destination address of an IPv6 packet.

When an ISATAP tunnel is used, the destination address of an IPv6 packet and the IPv6 address of a tunnel interface both adopt special ISATAP addresses. The ISATAP address format is prefix(64bit):0:5EFE:ip-address. The 64-bit prefix is the prefix of a valid IPv6 unicast address, but ip-address is a 32-bit source IPv4 address in the form of abcd:efgh, which need not be globally unique. Through the embedded IPv4 address, an ISATAP tunnel can automatically be created to transfer IPv6 packets.

The ISATAP tunnel is mainly used for connection between IPv6 routers or between a host and an IPv6 router over an IPv4 network.

Figure 76 Principle of ISATAP tunnel



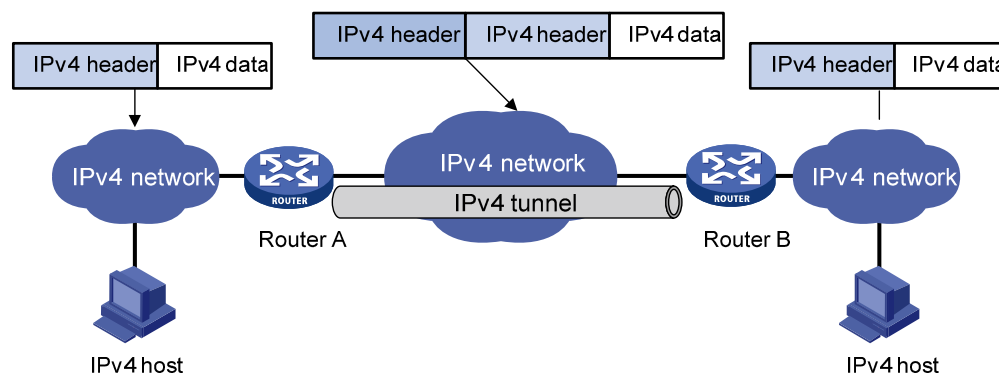
IPv4 over IPv4 tunnel

The IPv4 over IPv4 tunneling protocol (RFC 1853) is developed for IP data packet encapsulation so that data can be transferred from one IPv4 network to another IPv4 network.

Encapsulation and de-encapsulation

Packets traveling through a tunnel undergo an encapsulation and de-encapsulation process. [Figure 77](#) shows these two processes.

Figure 77 Principle of IPv4 over IPv4 tunnel



- Encapsulation

The encapsulation follows these steps.

1. The interface of Router A connecting to an IPv4 host receives an IP packet and submits it to the IP protocol stack for processing.
2. The IP protocol stack determines how to route the packet according to the destination address in the IP header. If the packet must be routed to the IPv4 host connected to Router B, the packet is sent to Router A's tunnel interface that is connected to Router B.
3. After the tunnel interface receives the packet, the packet is added with an outer IPv4 header and submitted to the IP protocol stack for processing. The IP protocol stack determines the outgoing interface of the tunnel according to the IP header.

- De-encapsulation

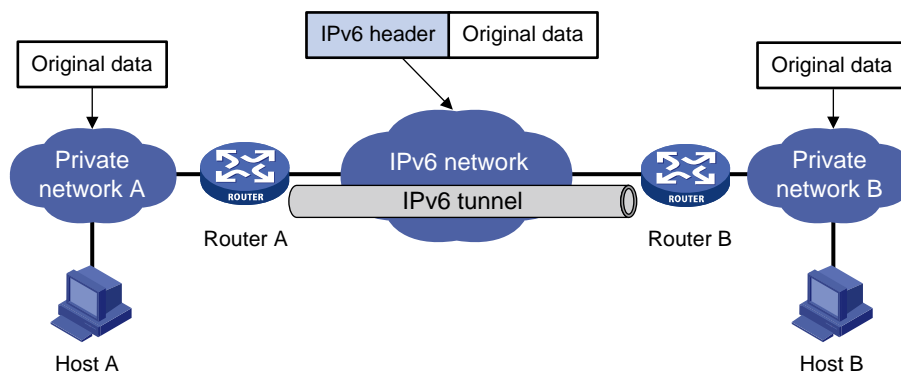
The de-encapsulation follows these steps.

1. The IP packet received from the IPv4 network interface is sent to the IP protocol stack, which then checks the protocol number in the IP header.
2. If the protocol number is IPv4, the IP packet is sent to the tunnel module for de-encapsulation.
3. The de-encapsulated IP packet is sent back to the IP protocol stack for processing.

IPv4/IPv6 over IPv6 tunnel

The IPv4/IPv6 over IPv6 tunneling protocol (RFC 2473) is developed for IPv4 or IPv6 data packet encapsulation so that encapsulated packets can be transmitted over an IPv6 network. The encapsulated packets are IPv6 tunnel packets.

Figure 78 Principle of IPv4/IPv6 over IPv6 tunnel



The original data in [Figure 78](#) refers to an IPv4 or IPv6 packet.

Encapsulation and de-encapsulation

The encapsulation follows these steps.

1. After receiving the original packet, the interface of Router A connecting private network A submits it to the corresponding data module for processing. The data module then determines how to route the packet.
2. If the packet must be routed to Host B connected to Router B, the packet is sent to Router A's tunnel interface that is connected to Router B.
3. After receiving the packet, the tunnel interface adds an IPv6 header to it and submits it to the IPv6 module for processing.
4. The IPv6 module re-determines a route according to the destination address in the IPv6 header.

The de-encapsulation follows these steps.

1. The packet received from the IPv6 network interface is sent to the IPv6 module for processing.
2. If the passenger protocol is IPv4 or IPv6, the packet is sent to the tunnel processing module for de-encapsulation.
3. The de-encapsulated packet is sent to the corresponding protocol module for the secondary routing process.

GRE tunnel

GRE is a protocol designed for encapsulating and carrying the packets of one network layer protocol (for example, IP or IPX) over another network layer protocol (for example, IP). GRE is a tunneling technology and serves as a Layer 3 tunneling protocol.

A GRE tunnel is a virtual point-to-point connection for transferring encapsulated packets. Packets are encapsulated at one end of the tunnel and de-encapsulated at the other end. [Figure 79](#) shows the encapsulation and de-encapsulation processes.

Figure 79 X protocol networks interconnected through the GRE tunnel



The following process takes the network shown in [Figure 79](#) as an example to describe how an X protocol packet traverses the IP network through a GRE tunnel.

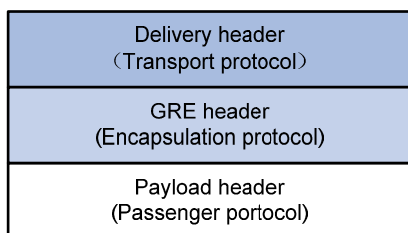
Encapsulation process

1. After receiving an X protocol packet through the interface connected to Group 1, Router A submits it to the X protocol for processing.
2. The X protocol checks the destination address field in the packet header to determine how to route the packet.
3. If the packet must be tunneled to reach its destination, Router A sends it to the tunnel interface.
4. Upon receipt of the packet, the tunnel interface encapsulates it in a GRE packet. Then, the system encapsulates the packet in an IP packet and forwards the IP packet based on its destination address and the routing table.

Format of an encapsulated packet

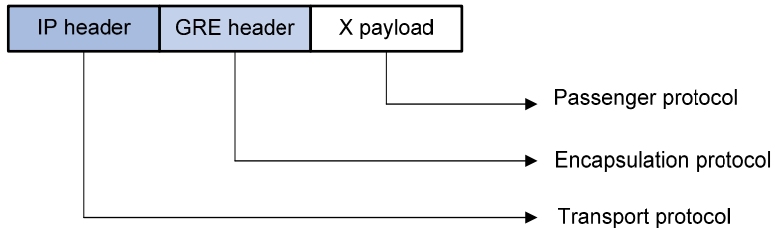
[Figure 80](#) shows the format of an encapsulated packet.

Figure 80 Format of an encapsulated packet



As an example, [Figure 81](#) shows the format of an X packet encapsulated for transmission over an IP tunnel.

Figure 81 Format of an X packet encapsulated for transmission over an IP tunnel



These terms are involved in the format:

- Payload: Packet that must be encapsulated and transmitted.
- Passenger protocol: Protocol that the payload packet uses, IPX in the example.
- Encapsulation or carrier protocol: Protocol used to encapsulate the payload packet, which is GRE.
- Delivery or transport protocol: Protocol used to encapsulate the GRE packet and then forward the packet to the other end of the tunnel, IP in this example.

Depending on the transport protocol, there are two tunnel modes, GRE over IPv4 and GRE over IPv6 tunnels.

De-encapsulation process

De-encapsulation is the reverse process of encapsulation and follows these steps.

1. Upon receiving an IP packet from the tunnel interface, Router B checks the destination address.
2. If the destination is itself, Router B strips off the IP header of the packet and submits the resulting packet to the GRE protocol.
3. The GRE protocol checks the key, checksum and sequence number in the packet, and then strips off the GRE header and submits the payload to the X protocol for forwarding.

Encapsulation and de-capsulation processes on both ends of the GRE tunnel and the resulting increase in data volumes degrades the forwarding efficiency of a GRE-enabled device to some extent.

Protocols and standards

- RFC 1853, *IP in IP Tunneling*
- RFC 2473, *Generic Packet Tunneling in IPv6 Specification*
- RFC 2893, *Transition Mechanisms for IPv6 Hosts and Routers*
- RFC 3056, *Connection of IPv6 Domains via IPv4 Clouds*
- RFC 4214, *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*

Configuration task list

| Task | Remarks | |
|--|---|--------------------|
| Configuring a tunnel interface | Required. | |
| Configuring an IPv6 over IPv4 tunnel | Configuring an IPv6 manual tunnel | Optional. |
| | Configuring a 6to4 tunnel | Use one as needed. |
| | Configuring an ISATAP tunnel | |
| Configuring an IPv4 over IPv4 tunnel | Optional. | |
| Configuring an IPv4 over IPv6 tunnel | Optional. | |
| Configuring an IPv6 over IPv6 tunnel | Optional. | |
| Configuring a GRE over IPv4 tunnel | Optional. | |
| Configuring a GRE over IPv6 tunnel | Optional. | |

Configuring a tunnel interface

Configuration prerequisites

Before configuring a tunnel interface on a switch, you may need create a service loopback group with its service type as Tunnel, and add unused Layer 2 Ethernet interfaces of the switch to the service loopback group. For more information about the service loopback group, see *Layer 2—LAN Switching Configuration Guide*.

Configuration procedure

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Create a tunnel interface and enter its view. | interface tunnel <i>number</i> | Required. By default, no tunnel interface is created. |
| 3. Configure the description for the interface. | description <i>text</i> | Optional. By default, the description of a tunnel interface is Tunnelnumber Interface . |
| 4. Reference a service loopback group. | service-loopback-group <i>number</i> | Required. By default, the tunnel does not reference any service loopback group. |

| Step | Command | Remarks |
|---|--|--|
| 5. Set the MTU of the tunnel interface. | mtu <i>mtu-size</i> | Optional. The MTU size of the tunnel interface cannot be larger than 1432 bytes. All tunnel interfaces created on the switch use the same MTU. |
| 6. Set the bandwidth of the tunnel interface. | tunnel bandwidth <i>bandwidth-value</i> | Optional. By default, the bandwidth of the tunnel interface is 64 kbps. |
| 7. Restore the default setting of the tunnel interface. | default | Optional. |
| 8. Shut down the tunnel interface. | shutdown | Optional. By default, the interface is up. |

Whether a packet can be sent successfully depends on the bandwidth of the tunnel interface and especially that of the output interface of the packet. As a result, you must refer to the bandwidth of the output interface when setting the bandwidth of the tunnel interface with **tunnel bandwidth**.

Before an IP unicast packet enters a tunnel, the system compares the length of the packet against the MTU size of the tunnel interface. If the packet length exceeds the tunnel MTU, the system fragments the IP unicast packet and then sends the packet fragments through the tunnel.

On an IRF virtual device comprising the HP 5800 and 5820X switches, add the unused Layer 2 Ethernet interfaces only on the HP 5820X to the service loopback group.

By default, the switch is disabled from sending ICMP destination unreachable messages. To enable the switch to send ICMP destination unreachable messages, use **ip unreachable enable**.

On an HP 5800 switch, when the MTU of an IP multicast forwarded by the tunnel interface is larger than the specified MTU, the switch returns an ICMP error packet, notifying the source device to adjust the MTU. The adjusted MTU of the IP multicast must be no larger than the specified one.

To forward an encapsulated packet a second time at Layer 3, a switch does not use the destination address or routing table, but sends the packet to the loopback interface. The loopback interface then returns the packet to the forwarding module for Layer 3 forwarding. Therefore, you must assign the tunnel interface to an existing service loopback group. Otherwise, the tunnel interface cannot be up and packets cannot be transmitted over the tunnel. For more information about the service loopback group, see *Layer 2—LAN Switching Configuration Guide*.

Configuring an IPv6 manual tunnel

Configuration prerequisites

Configure IP addresses for interfaces (such as the VLAN interface, Layer 3 Ethernet interface, and loopback interface) on the device to ensure normal communication. One of the interfaces is used as the source interface of the tunnel.

Configuration procedure

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enable IPv6. | ipv6 | Required. By default, the IPv6 packet forwarding function is disabled. |
| 3. Enter tunnel interface view. | interface tunnel <i>number</i> | — |
| 4. Configure an IPv6 address for the tunnel interface. | Configure a global unicast IPv6 address or a site-local address. ipv6 address { ipv6-address prefix-length ipv6-address/prefix-length } | Required. Use either command. |
| | Configure a link-local IPv6 address. ipv6 address ipv6-address/prefix-length eui-64 | By default, no IPv6 global unicast address or site-local address is configured for the tunnel interface. |
| | ipv6 address auto link-local ipv6 address <i>ipv6-address</i> link-local | Optional. By default, a link-local address is automatically created when an IPv6 global unicast address or site-local address is configured. |
| 5. Specify the IPv6 manual tunnel mode. | tunnel-protocol ipv6-ipv4 | Required. By default, the tunnel is a GRE over IPv4 tunnel. The same tunnel mode should be configured at both ends of the tunnel. Otherwise, packet delivery fails. |
| 6. Configure a source address or interface for the tunnel. | source { <i>ip-address</i> <i>interface-type interface-number</i> } | Required. By default, no source address or interface is configured for the tunnel. |
| 7. Configure a destination address for the tunnel. | destination <i>ip-address</i> | Required. By default, no destination address is configured for the tunnel. |

After a tunnel interface is deleted, all features configured on the tunnel interface is deleted.

To encapsulate and forward IPv6 packets whose destination address does not belong to the subnet where the current tunnel interface resides, you must configure a static route or dynamic routing for forwarding those packets through this tunnel interface.

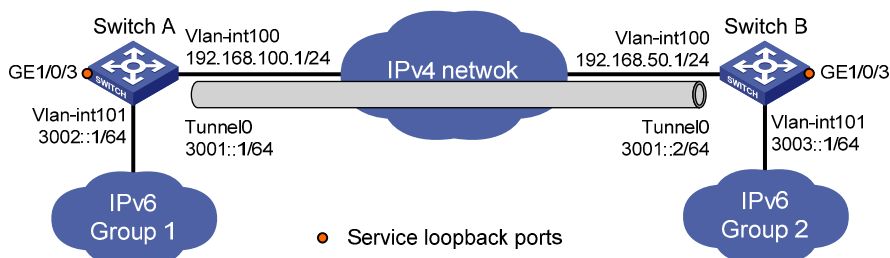
- If you configure a static route to that destination IPv6 address, specify this tunnel interface as the outbound interface, or the peer tunnel interface address as the next hop. A similar configuration must be performed at the other tunnel end.
- If you configure dynamic routing at both ends, enable the dynamic routing protocol on both tunnel interfaces. For the detailed configuration, see *Layer 3—IP Routing Configuration Guide*.

Configuration example

Network requirements

As shown in Figure 82, two IPv6 networks are connected to an IPv4 network through Switch A and Switch B respectively. Configure an IPv6 manual tunnel between Switch A and Switch B to make the two IPv6 networks reachable to each other.

Figure 82 Network diagram for an IPv6 manual tunnel



Configuration procedure

Make sure that Switch A and Switch B have the corresponding VLAN interfaces created and are reachable to each other.

- Configuration on Switch A

Enable IPv6.

```
<SwitchA> system-view
```

```
[SwitchA] ipv6
```

Configure an IPv4 address for VLAN-interface 100.

```
[SwitchA] interface vlan-interface 100
```

```
[SwitchA-Vlan-interface100] ip address 192.168.100.1 255.255.255.0
```

```
[SwitchA-Vlan-interface100] quit
```

Configure an IPv6 address for VLAN-interface 101.

```
[SwitchA] interface vlan-interface 101
```

```
[SwitchA-Vlan-interface101] ipv6 address 3002::1 64
```

```
[SwitchA-Vlan-interface101] quit
```

Configure a manual IPv6 tunnel.

```
[SwitchA] interface tunnel 0
[SwitchA-Tunnel0] ipv6 address 3001::1/64
[SwitchA-Tunnel0] source vlan-interface 100
[SwitchA-Tunnel0] destination 192.168.50.1
[SwitchA-Tunnel0] tunnel-protocol ipv6-ipv4
[SwitchA-Tunnel0] quit
```

Create service loopback group 1 to support the tunnel service.

```
[SwitchA] service-loopback group 1 type tunnel
```

Add GigabitEthernet1/0/3 to service loopback group 1.

```
[SwitchA] interface GigabitEthernet1/0/3
[SwitchA-GigabitEthernet1/0/3] undo stp enable
[SwitchA-GigabitEthernet1/0/3] undo ndp enable
[SwitchA-GigabitEthernet1/0/3] undo lldp enable
[SwitchA-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchA-GigabitEthernet1/0/3] quit
```

Reference service loopback group 1 on the tunnel.

```
[SwitchA] interface tunnel 0
[SwitchA-Tunnel0] service-loopback-group 1
[SwitchA-Tunnel0] quit
```

Configure a static route to IPv6 Group 2 through Tunnel 0 on Switch A.

```
[SwitchA] ipv6 route-static 3003:: 64 tunnel 0
```

- **Configuration on Switch B**

Enable IPv6.

```
<SwitchB> system-view
[SwitchB] ipv6
```

Configure an IPv4 address for VLAN-interface 100.

```
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ip address 192.168.50.1 255.255.255.0
[SwitchB-Vlan-interface100] quit
```

Configure an IPv6 address for VLAN-interface 101.

```
[SwitchB] interface vlan-interface 101
[SwitchB-Vlan-interface101] ipv6 address 3003::1 64
[SwitchB-Vlan-interface101] quit
```

Configure an IPv6 manual tunnel.

```
[SwitchB] interface tunnel 0
[SwitchB-Tunnel0] ipv6 address 3001::2/64
[SwitchB-Tunnel0] source vlan-interface 100
[SwitchB-Tunnel0] destination 192.168.100.1
[SwitchB-Tunnel0] tunnel-protocol ipv6-ipv4
[SwitchB-Tunnel0] quit
```

Create service loopback group 1 to support the tunnel service.

```
[SwitchB] service-loopback group 1 type tunnel
```

Add GigabitEthernet1/0/3 to service loopback group 1.


```

[SwitchB] interface GigabitEthernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] undo stp enable
[SwitchB-GigabitEthernet1/0/3] undo ndp enable
[SwitchB-GigabitEthernet1/0/3] undo lldp enable
[SwitchB-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchB-GigabitEthernet1/0/3] quit

# Reference service loopback group 1 on the tunnel.
[SwitchB] interface tunnel 0
[SwitchB-Tunnel0] service-loopback-group 1
[SwitchB-Tunnel0] quit

# Configure a static route to IPv6 Group 1 through Tunnel 0 on Switch B.
[SwitchB] ipv6 route-static 3002:: 64 tunnel 0

```

Configuration verification

After the above configurations, display the status of the tunnel interfaces on Switch A and Switch B, respectively.

```

[SwitchA] display ipv6 interface tunnel 0 verbose
Tunnel0 current state :UP
Line protocol current state :UP
IPv6 is enabled, link-local address is FE80::C0A8:6401
  Global unicast address(es):
    3001::1, subnet is 3001::/64
  Joined group address(es):
    FF02::1:FF00:0
    FF02::1:FF00:1
    FF02::1:FFA8:6401
    FF02::2
    FF02::1
  MTU is 1480 bytes
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
  InReceives:                55
...
[SwitchB] display ipv6 interface tunnel 0 verbose
Tunnel0 current state :UP
Line protocol current state :UP
IPv6 is enabled, link-local address is FE80::C0A8:3201
  Global unicast address(es):
    3001::2, subnet is 3001::/64
  Joined group address(es):
    FF02::1:FF00:0
    FF02::1:FF00:1
    FF02::1:FFA8:3201
    FF02::2
    FF02::1
  MTU is 1480 bytes

```

```

ND reachable time is 30000 milliseconds
ND retransmit interval is 1000 milliseconds
Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
  InReceives:                55
...
# Ping the IPv6 address of VLAN-interface 101 at the peer end from Switch A.
[SwitchA] ping ipv6 3003::1
PING 3003::1 : 56 data bytes, press CTRL_C to break
  Reply from 3003::1
  bytes=56 Sequence=1 hop limit=64 time = 1 ms
  Reply from 3003::1
  bytes=56 Sequence=2 hop limit=64 time = 1 ms
  Reply from 3003::1
  bytes=56 Sequence=3 hop limit=64 time = 1 ms
  Reply from 3003::1
  bytes=56 Sequence=4 hop limit=64 time = 1 ms
  Reply from 3003::1
  bytes=56 Sequence=5 hop limit=64 time = 1 ms

--- 3003::1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 1/1/1 ms

```

Configuring a 6to4 tunnel

Configuration prerequisites

Configure IP addresses for interfaces (such as the VLAN interface, Layer 3 Ethernet interface, and loopback interface) on the device to ensure normal communication. One of the interfaces is used as the source interface of the tunnel.

Configuration procedure

| Step | Command | Remarks | |
|--|---|---|---|
| 1. Enter system view. | system-view | — | |
| 2. Enable IPv6. | ipv6 | Required. By default, the IPv6 packet forwarding function is disabled. | |
| 3. Enter tunnel interface view. | interface tunnel <i>number</i> | — | |
| 4. Configure an IPv6 address for the tunnel interface. | Configure an IPv6 global unicast address or a site-local address. | ipv6 address { ipv6-address prefix-length ipv6-address/prefix-length } | Required. Use either command. |
| | Configure an IPv6 link-local address. | ipv6 address ipv6-address/prefix-length eui-64 | By default, no IPv6 global unicast address or site-local address is configured for the tunnel interface. |
| | | ipv6 address auto link-local | Optional. By default, a link-local address is automatically generated when an IPv6 global unicast address or site-local address is configured. |
| 5. Specify the 6to4 tunnel mode. | tunnel-protocol ipv6-ipv4 6to4 | Required. By default, the tunnel is a GRE over IPv4 tunnel. The same tunnel mode should be configured at both ends of the tunnel. Otherwise, packet delivery fails. | |
| 6. Configure a source address or interface for the tunnel. | source { ip-address interface-type interface-number } | Required. By default, no source address or interface is configured for the tunnel. | |

No destination address needs to be configured for a 6to4 tunnel because the destination address can automatically be obtained from the IPv4 address embedded in the 6to4 IPv6 address.

To encapsulate and forward IPv6 packets whose destination address does not belong to the subnet where the receiving tunnel interface resides, configure a static route to reach the destination IPv6 address through this tunnel interface on the device. Because automatic tunnels do not support dynamic routing, configure a static route to that destination IPv6 address with this tunnel interface as the outbound interface or the peer tunnel interface address as the next hop. A similar configuration needs to be performed at the other tunnel end. For the detailed configuration, see *Layer 3—IP Routing Configuration Guide*.

The automatic tunnel interfaces using the same encapsulation protocol cannot share the same source IP address.

6to4 tunnel configuration example

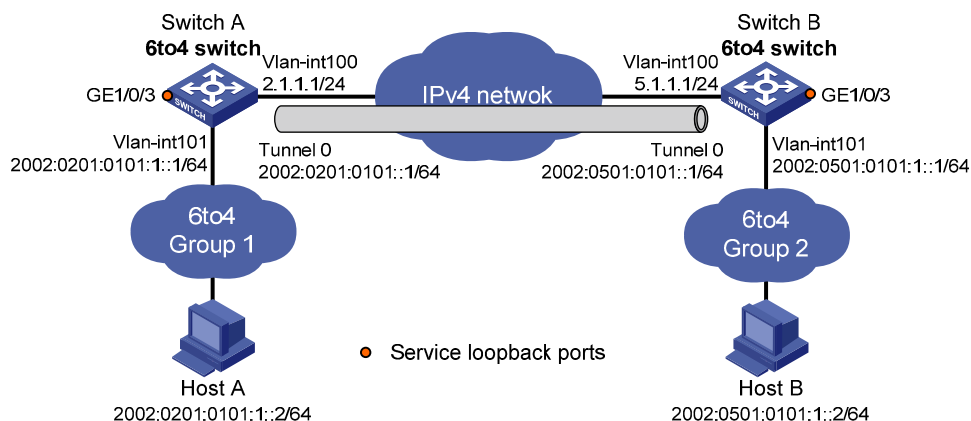
Network requirements

As shown in Figure 83, two 6to4 networks are connected to an IPv4 network through two 6to4 switches (Switch A and Switch B) respectively. Configure a 6to4 tunnel to make Host A and Host B reachable to each other.

To enable communication between 6to4 networks, configure 6to4 addresses for 6to4 switches and hosts in the 6to4 networks.

- The IPv4 address of VLAN-interface 100 on Switch A is 2.1.1.1/24, and the corresponding 6to4 prefix is 2002:0201:0101::/48 after it is translated to an IPv6 address. Assign interface Tunnel 0 to subnet 2002:0201:0101::/64 and VLAN-interface 101 to subnet 2002:0201:0101:1::/64.
- The IPv4 address of VLAN-interface 100 on Switch B is 5.1.1.1/24, and the corresponding 6to4 prefix is 2002:0501:0101::/48 after it is translated to an IPv6 address. Assign interface Tunnel 0 to subnet 2002:0501:0101::/64 and VLAN-interface 101 to subnet 2002:0501:0101:1::/64.

Figure 83 Network diagram for a 6to4 tunnel



Configuration procedure

Make sure that Switch A and Switch B have the corresponding VLAN interfaces created and are reachable to each other.

- Configuration on Switch A

Enable IPv6.

```
<SwitchA> system-view
[SwitchA] ipv6
```

Configure an IPv4 address for VLAN-interface 100.

```
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ip address 2.1.1.1 24
[SwitchA-Vlan-interface100] quit
```

Configure an IPv6 address for VLAN-interface 101.

```
[SwitchA] interface vlan-interface 101
[SwitchA-Vlan-interface101] ipv6 address 2002:0201:0101:1::1/64
[SwitchA-Vlan-interface101] quit
```

Configure a 6to4 tunnel.

```
[SwitchA] interface tunnel 0
[SwitchA-Tunnel0] ipv6 address 2002:201:101::1/64
[SwitchA-Tunnel0] source vlan-interface 100
[SwitchA-Tunnel0] tunnel-protocol ipv6-ipv4 6to4
[SwitchA-Tunnel0] quit
```

Create service loopback group 1 to support the tunnel service.

```
[SwitchA] service-loopback group 1 type tunnel
```

Add GigabitEthernet1/0/3 to service loopback group 1.

```
[SwitchA] interface GigabitEthernet 1/0/3
[SwitchA-GigabitEthernet1/0/3] undo stp enable
[SwitchA-GigabitEthernet1/0/3] undo ndp enable
[SwitchA-GigabitEthernet1/0/3] undo lldp enable
[SwitchA-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchA-GigabitEthernet1/0/3] quit
```

Reference service loopback group 1 on the tunnel.

```
[SwitchA] interface tunnel 0
[SwitchA-Tunnel0] service-loopback-group 1
[SwitchA-Tunnel0] quit
```

Configure a static route whose destination address is 2002::/16 and next-hop is the tunnel interface.

```
[SwitchA] ipv6 route-static 2002:: 16 tunnel 0
```

- **Configuration on Switch B**

Enable IPv6.

```
<SwitchB> system-view
[SwitchB] ipv6
```

Configure an IPv4 address for VLAN-interface 100.

```
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ip address 5.1.1.1 24
[SwitchB-Vlan-interface100] quit
```

Configure an IPv6 address for VLAN-interface 101.

```
[SwitchB] interface vlan-interface 101
[SwitchB-Vlan-interface101] ipv6 address 2002:0501:0101:1::1/64
[SwitchB-Vlan-interface101] quit
```

Configure the 6to4 tunnel.

```
[SwitchB] interface tunnel 0
[SwitchB-Tunnel0] ipv6 address 2002:0501:0101::1/64
[SwitchB-Tunnel0] source vlan-interface 100
[SwitchB-Tunnel0] tunnel-protocol ipv6-ipv4 6to4
[SwitchB-Tunnel0] quit
```

Create service loopback group 1 to support the tunnel service.

```
[SwitchB] service-loopback group 1 type tunnel
```

```

# Add GigabitEthernet1/0/3 to service loopback group 1.
[SwitchB] interface GigabitEthernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] undo stp enable
[SwitchB-GigabitEthernet1/0/3] undo ndp enable
[SwitchB-GigabitEthernet1/0/3] undo lldp enable
[SwitchB-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchB-GigabitEthernet1/0/3] quit

# Reference service loopback group 1 on the tunnel.
[SwitchB] interface tunnel 0
[SwitchB-Tunnel0] service-loopback-group 1
[SwitchB-Tunnel0] quit

# Configure a static route whose destination address is 2002::/16 and the next hop is the tunnel
interface.
[SwitchB] ipv6 route-static 2002:: 16 tunnel 0

```

Configuration verification

After the above configuration, ping Host B from Host A or ping Host A from Host B.

```
D:\>ping6 -s 2002:201:101:1::2 2002:501:101:1::2
```

```
Pinging 2002:501:101:1::2
```

```
from 2002:201:101:1::2 with 32 bytes of data:
```

```
Reply from 2002:501:101:1::2: bytes=32 time=13ms
```

```
Reply from 2002:501:101:1::2: bytes=32 time=1ms
```

```
Reply from 2002:501:101:1::2: bytes=32 time=1ms
```

```
Reply from 2002:501:101:1::2: bytes=32 time<1ms
```

```
Ping statistics for 2002:501:101:1::2:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 13ms, Average = 3ms
```

Configuring an ISATAP tunnel

Configuration prerequisites

Configure IP addresses for interfaces (such as the VLAN interface, and loopback interface) on the device to ensure normal communication. One of the interfaces is used as the source interface of the tunnel.

Configuration procedure

| Step | Command | Remarks |
|--|--|---|
| 1. Enter system view. | system-view | — |
| 2. Enable IPv6. | ipv6 | Required. By default, the IPv6 forwarding function is disabled. |
| 3. Enter tunnel interface view. | interface tunnel <i>number</i> | — |
| 4. Configure an IPv6 address for the tunnel interface. | Configure an IPv6 global unicast address or site-local address. ipv6 address { ipv6-address prefix-length ipv6-address/prefix-length } | Required. Use either command. |
| | Configure an IPv6 link-local address. ipv6 address ipv6-address/prefix-length eui-64 | By default, no IPv6 global unicast address is configured for the tunnel interface. |
| | ipv6 address auto link-local | Optional. By default, a link-local address is automatically generated when an IPv6 global unicast address or link-local address is configured. |
| 5. Specify the ISATAP tunnel mode. | tunnel-protocol ipv6-ipv4 isatap | Required. By default, the tunnel is a GRE over IPv4 tunnel. The same tunnel mode should be configured at both ends of the tunnel. Otherwise, packet delivery fails. |
| 6. Configure a source address or interface for the tunnel. | source { ip-address interface-type interface-number } | Required. By default, no source address or interface is configured for the tunnel. |

No destination address needs to be configured for an ISATAP tunnel. The destination address of the tunnel can be automatically obtained through the IPv4 address embedded in the ISATAP address.

To encapsulate and forward IPv6 packets whose destination address does not belong to the subnet where the receiving tunnel interface resides, configure a static route to reach the destination IPv6 address through this tunnel interface on the device. Because automatic tunnels do not support dynamic routing, configure a static route to that destination IPv6 address with this tunnel interface as the outbound interface or the peer tunnel interface address as the next hop. A similar configuration needs to be performed at the other tunnel end. For the detailed configuration, see *Layer 3—IP Routing Configuration Guide*.

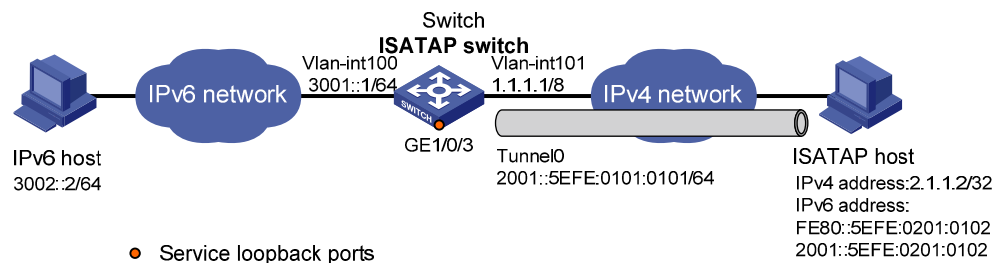
The automatic tunnel interfaces using the same encapsulation protocol cannot share the same source IP address.

Configuration example

Network requirements

As shown in Figure 84, an IPv6 network is connected to an IPv4 network through an ISATAP switch. The destination address of the tunnel is an ISATAP address. It is required that IPv6 hosts in the IPv4 network can access the IPv6 network through the ISATAP tunnel.

Figure 84 Network diagram for an ISATAP tunnel



Configuration procedure

Make sure that the corresponding VLAN interfaces have been created on the switch.

Make sure that VLAN-interface 101 on the ISATAP switch and the ISATAP host are reachable to each other.

- Configuration on the switch

Enable IPv6.

```
<Switch> system-view  
[Switch] ipv6
```

Configure addresses for interfaces.

```
[Switch] interface vlan-interface 100  
[Switch-Vlan-interface100] ipv6 address 3001::1/64  
[Switch-Vlan-interface100] quit  
[Switch] interface vlan-interface 101  
[Switch-Vlan-interface101] ip address 1.1.1.1 255.0.0.0  
[Switch-Vlan-interface101] quit
```

Configure an ISATAP tunnel.

```
[Switch] interface tunnel 0  
[Switch-Tunnel0] ipv6 address 2001::5efe:0101:0101 64  
[Switch-Tunnel0] source vlan-interface 101  
[Switch-Tunnel0] tunnel-protocol ipv6-ipv4 isatap
```

Disable the RA suppression so that hosts can acquire information such as the address prefix from the RA message released by the ISATAP switch.

```
[Switch-Tunnel0] undo ipv6 nd ra halt  
[Switch-Tunnel0] quit
```

Create service loopback group 1 to support the tunnel service.

```
[Switch] service-loopback group 1 type tunnel
```



```
# Add GigabitEthernet1/0/3 to service loopback group 1.
[Switch] interface GigabitEthernet 1/0/3
[Switch-GigabitEthernet1/0/3] undo stp enable
[Switch-GigabitEthernet1/0/3] undo ndp enable
[Switch-GigabitEthernet1/0/3] undo lldp enable
[Switch-GigabitEthernet1/0/3] port service-loopback group 1
[Switch-GigabitEthernet1/0/3] quit
```

```
# Reference service loopback group 1 on the tunnel.
```

```
[Switch] interface tunnel 0
[Switch-Tunnel0] service-loopback-group 1
[Switch-Tunnel0] quit
```

```
# Configure a static route to the ISATAP host.
```

```
[Switch] ipv6 route-static 2001:: 16 tunnel 0
```

- Configuration on the ISATAP host

The specific configuration on the ISATAP host is related to its operating system. The following example shows the configuration of the host running the Windows XP.

```
# Install IPv6.
```

```
C:\>ipv6 install
```

```
# On a Windows XP-based host, the ISATAP interface is usually interface 2. Configure the IPv4 address of the ISATAP router on the interface to complete the configuration on the host. Before doing that, display the ISATAP interface information:
```

```
C:\>ipv6 if 2
Interface 2: Automatic Tunneling Pseudo-Interface
  Guid {48FCE3FC-EC30-E50E-F1A7-71172AEEE3AE}
  does not use Neighbor Discovery
  does not use Router Discovery
  routing preference 1
  EUI-64 embedded IPv4 address: 0.0.0.0
  router link-layer address: 0.0.0.0
  preferred link-local fe80::5efe:2.1.1.2, life infinite
  link MTU 1280 (true link MTU 65515)
  current hop limit 128
  reachable time 42500ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
  default site prefix length 48
```

```
# A link-local address (fe80::5efe:2.1.1.2) in the ISATAP format was automatically generated for the ISATAP interface. Configure the IPv4 address of the ISATAP switch on the ISATAP interface.
```

```
C:\>ipv6 rlu 2 1.1.1.1
```

After carrying out the above command, look at the information on the ISATAP interface.

```
C:\>ipv6 if 2
```

```
Interface 2: Automatic Tunneling Pseudo-Interface
  Guid {48FCE3FC-EC30-E50E-F1A7-71172AEEEE3AE}
  does not use Neighbor Discovery
  uses Router Discovery
  routing preference 1
  EUI-64 embedded IPv4 address: 2.1.1.2
  router link-layer address: 1.1.1.1
    preferred global 2001::5efe:2.1.1.2, life 29d23h59m46s/6d23h59m46s (public)
    preferred link-local fe80::5efe:2.1.1.2, life infinite
  link MTU 1500 (true link MTU 65515)
  current hop limit 255
  reachable time 42500ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
  default site prefix length 48
```

By comparison, it is found that the host acquires the address prefix 2001::/64 and automatically generates the address 2001::5efe:2.1.1.2. Meanwhile, "uses Router Discovery" is displayed, indicating that the router discovery function is enabled on the host. Ping the IPv6 address of the tunnel interface of the switch. If the address is successfully pinged, an ISATAP tunnel is established.

```
C:\>ping 2001::5efe:1.1.1.1
```

```
Pinging 2001::5efe:1.1.1.1 with 32 bytes of data:
```

```
Reply from 2001::5efe:1.1.1.1: time=1ms
Reply from 2001::5efe:1.1.1.1: time=1ms
Reply from 2001::5efe:1.1.1.1: time=1ms
Reply from 2001::5efe:1.1.1.1: time=1ms
```

```
Ping statistics for 2001::5efe:1.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Configuration verification

After the above configurations, the ISATAP host can access the host in the IPV6 network.

Configuring an IPv4 over IPv4 tunnel

Configuration prerequisites

Configure IP addresses for interfaces (such as the VLAN interface, and loopback interface) on the device to ensure normal communication. One of the interfaces is used as the source interface of the tunnel.

Configuration procedure

| Step | Command | Remarks |
|--|---|---|
| 1. Enter system view. | system-view | — |
| 2. Enter tunnel interface view. | interface tunnel <i>number</i> | — |
| 3. Configure an IPv4 address for the tunnel interface. | ip address <i>ip-address</i> { <i>mask</i> <i>mask-length</i> } [sub] | Required. By default, no IPv4 address is configured for the tunnel interface. |
| 4. Specify the IPv4 over IPv4 tunnel mode. | tunnel-protocol ipv4-ipv4 | Optional. By default, the tunnel is a GRE over IPv4 tunnel. The same tunnel mode should be configured at both ends of the tunnel. Otherwise, packet delivery fails. |
| 5. Configure a source address or interface for the tunnel interface. | source { <i>ip-address</i> <i>interface-type interface-number</i> } | Required. By default, no source address or interface is configured for the tunnel. |
| 6. Configure a destination address for the tunnel interface. | destination <i>ip-address</i> | Required. By default, no destination address is configured for the tunnel. |

To encapsulate and forward IPv4 packets whose destination address does not belong to the subnet where the receiving tunnel interface resides, you must configure a static route or dynamic routing for forwarding those packets through this tunnel interface.

If you configure a static route to that destination IPv4 address, specify this tunnel interface as the outbound interface, or the peer tunnel interface address as the next hop. A similar configuration needs to be performed at the other tunnel end. If you configure dynamic routing at both ends, enable the dynamic routing protocol on both tunnel interfaces. For the detailed configuration, see *Layer 3—IP Routing Configuration Guide*.

The IPv4 address of the local tunnel interface cannot be on the same subnet as the destination address of the tunnel.

The destination address of a route with a tunnel interface as the egress interface must not be on the same subnet as the destination address of the tunnel.

Two or more tunnel interfaces using the same encapsulation protocol must have different source and destination addresses.

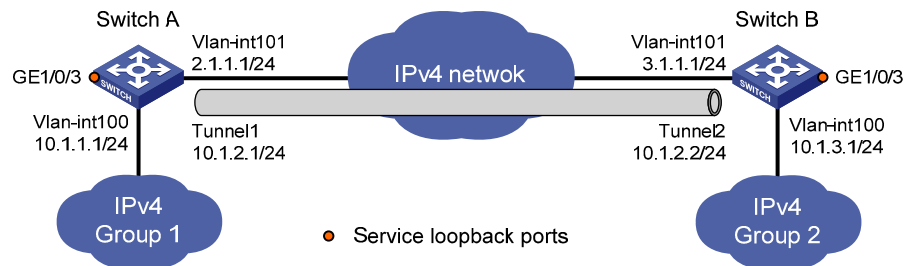
If you specify a source interface instead of a source address for the tunnel, the source address of the tunnel is the primary IP address of the source interface.

Configuration example

Network requirements

As shown in [Figure 85](#), the two subnets Group 1 and Group 2 use private IPv4 addresses. Configure an IPv4 over IPv4 tunnel between Switch A and Switch B to make the two subnets reachable to each other.

Figure 85 Network diagram for an IPv4 over IPv4 tunnel



Configuration procedure

Make sure that Switch A and Switch B have the corresponding VLAN interfaces created and are reachable to each other.

- Configuration on Switch A

Configure an IPv4 address for VLAN-interface 100.

```
<SwitchA> system-view
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ip address 10.1.1.1 255.255.255.0
[SwitchA-Vlan-interface100] quit
```

Configure an IPv4 address for VLAN-interface 101.

```
[SwitchA] interface vlan-interface 101
[SwitchA-Vlan-interface101] ip address 2.1.1.1 255.255.255.0
[SwitchA-Vlan-interface101] quit
```

Create the interface tunnel 1.

```
[SwitchA] interface tunnel 1
```

Configure an IPv4 address for the interface tunnel 1.

```
[SwitchA-Tunnel1] ip address 10.1.2.1 255.255.255.0
```

Configure the tunnel encapsulation mode.

```
[SwitchA-Tunnel1] tunnel-protocol ipv4-ipv4
```

Configure a source address for the interface Tunnel 1 (IP address of VLAN-interface 101).

```
[SwitchA-Tunnel1] source 2.1.1.1
```

Configure a destination address for the interface Tunnel 1 (IP address of VLAN-interface 101 of Switch B).

```
[SwitchA-Tunnel1] destination 3.1.1.1  
[SwitchA-Tunnel1] quit
```

Create service loopback group 1 to support the tunnel service.

```
[SwitchA] service-loopback group 1 type tunnel
```

Add GigabitEthernet1/0/3 to service loopback group 1.

```
[SwitchA] interface GigabitEthernet 1/0/3  
[SwitchA-GigabitEthernet1/0/3] undo stp enable  
[SwitchA-GigabitEthernet1/0/3] undo ndp enable  
[SwitchA-GigabitEthernet1/0/3] undo lldp enable  
[SwitchA-GigabitEthernet1/0/3] port service-loopback group 1  
[SwitchA-GigabitEthernet1/0/3] quit
```

Reference service loopback group 1 on the tunnel.

```
[SwitchA] interface tunnel 1  
[SwitchA-Tunnel1] service-loopback-group 1  
[SwitchA-Tunnel1] quit
```

Configure a static route from Switch through the interface Tunnel 1 to Group 2.

```
[SwitchA] ip route-static 10.1.3.0 255.255.255.0 tunnel 1
```

- Configuration on Switch B

Configure an IPv4 address for VLAN-interface 100.

```
<SwitchB> system-view  
[SwitchB] interface vlan-interface 100  
[SwitchB-Vlan-interface100] ip address 10.1.3.1 255.255.255.0  
[SwitchB-Vlan-interface100] quit
```

Configure an IPv4 address for VLAN-interface 101.

```
[SwitchB] interface vlan-interface 101  
[SwitchB-Vlan-interface101] ip address 3.1.1.1 255.255.255.0  
[SwitchB-Vlan-interface101] quit
```

Create the interface Tunnel 2.

```
[SwitchB] interface tunnel 2
```

Configure an IPv4 address for the interface Tunnel 2.

```
[SwitchB-Tunnel2] ip address 10.1.2.2 255.255.255.0
```

Configure the tunnel encapsulation mode.

```
[SwitchB-Tunnel2] tunnel-protocol ipv4-ipv4
```

Configure the source address for the interface Tunnel 2 (IP address of VLAN-interface 101).

```
[SwitchB-Tunnel2] source 3.1.1.1
```

Configure the destination address for the interface Tunnel 2 (IP address of VLAN-interface 101 of Switch A).

```
[SwitchB-Tunnel2] destination 2.1.1.1  
[SwitchB-Tunnel2] quit
```

```

# Create service loopback group 1 to support the tunnel service.
[SwitchB] service-loopback group 1 type tunnel

# Add GigabitEthernet1/0/3 to service loopback group 1.
[SwitchB] interface GigabitEthernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] undo stp enable
[SwitchB-GigabitEthernet1/0/3] undo ndp enable
[SwitchB-GigabitEthernet1/0/3] undo lldp enable
[SwitchB-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchB-GigabitEthernet1/0/3] quit

# Reference service loopback group 1 on the tunnel.
[SwitchB] interface tunnel 2
[SwitchB-Tunnel2] service-loopback-group 1
[SwitchB-Tunnel2] quit

# Configure a static route from Switch B through the interface Tunnel 2 to Group 1.
[SwitchB] ip route-static 10.1.1.0 255.255.255.0 tunnel 2

```

Configuration verification

After the above configuration, display the status of the tunnel interfaces on Switch A and Switch B:

```

[SwitchA] display interface tunnel 1
Tunnell current state: UP
Line protocol current state: UP
Description: Tunnell Interface
The Maximum Transmit Unit is 1480
Internet Address is 10.1.2.1/24 Primary
Encapsulation is TUNNEL, service-loopback-group ID is 1.
Tunnel source 2.1.1.1, destination 3.1.1.1
Tunnel protocol/transport IP/IP
    Last 300 seconds input:  0 bytes/sec, 0 packets/sec
    Last 300 seconds output: 2 bytes/sec, 0 packets/sec
    4 packets input, 256 bytes
    0 input error
    12 packets output, 768 bytes
    0 output error

[SwitchB] display interface tunnel 2
Tunnel2 current state: UP
Line protocol current state: UP
Description: Tunnel2 Interface
The Maximum Transmit Unit is 1480
Internet Address is 10.1.2.2/24 Primary
Encapsulation is TUNNEL, service-loopback-group ID is 1.
Tunnel source 3.1.1.1, destination 2.1.1.1
Tunnel protocol/transport IP/IP
    Last 300 seconds input:  0 bytes/sec, 0 packets/sec
    Last 300 seconds output: 0 bytes/sec, 0 packets/sec
    5 packets input, 320 bytes
    0 input error

```

```

    9 packets output,  576 bytes
    0 output error

# Ping the IPv4 address of the peer interface VLAN-interface 100 from Switch A.
[SwitchA] ping 10.1.3.1
  PING 10.1.3.1: 56  data bytes, press CTRL_C to break
    Reply from 10.1.3.1: bytes=56 Sequence=1 ttl=255 time=15 ms
    Reply from 10.1.3.1: bytes=56 Sequence=2 ttl=255 time=15 ms
    Reply from 10.1.3.1: bytes=56 Sequence=3 ttl=255 time=16 ms
    Reply from 10.1.3.1: bytes=56 Sequence=4 ttl=255 time=16 ms
    Reply from 10.1.3.1: bytes=56 Sequence=5 ttl=255 time=15 ms

--- 10.1.3.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 15/15/16 ms

```

Configuring an IPv4 over IPv6 tunnel

Configuration prerequisites

Configure IP addresses for interfaces (such as the VLAN interface, and loopback interface) on the device to ensure normal communication. One of the interfaces is used as the source interface of the tunnel.

Configuration procedure

| Step | Command | Remarks |
|--|--|---|
| 1. Enter system view. | system-view | — |
| 2. Enable IPv6. | ipv6 | Required. By default, the IPv6 packet forwarding function is disabled. |
| 3. Enter tunnel interface view. | interface tunnel <i>number</i> | — |
| 4. Configure an IPv4 address for the tunnel interface. | ip address ip-address { mask mask-length } [sub] | Required. By default, no IPv4 address is configured for the tunnel interface. |
| 5. Specify the IPv4 over IPv6 tunnel mode. | tunnel-protocol ipv4-ipv6 | Optional. By default, the tunnel is a GRE over IPv4 tunnel. The same tunnel mode should be configured at both ends of the tunnel. Otherwise, packet delivery fails. |
| 6. Configure the source address or interface for the tunnel interface. | source { ipv6-address interface-type interface-number } | Required. By default, no source address or interface is configured for the tunnel. |

| Step | Command | Remarks |
|--|--|---|
| 7. Configure the destination address for the tunnel interface. | destination <i>ipv6-address</i> | Required. By default, no destination address is configured for the tunnel. |

To encapsulate and forward IPv4 packets whose destination address does not belong to the subnet where the receiving tunnel interface resides, configure a static route or dynamic routing for forwarding those packets through this tunnel interface.

If you configure a static route to that destination IPv4 address, specify this tunnel interface as the outbound interface, or the peer tunnel interface address as the next hop. A similar configuration needs to be performed at the other tunnel end. If you configure dynamic routing at both ends, enable the dynamic routing protocol on both tunnel interfaces. For the detailed configuration, see *Layer 3—IP Routing Configuration Guide*.

Two or more tunnel interfaces using the same encapsulation protocol must have different source and destination addresses.

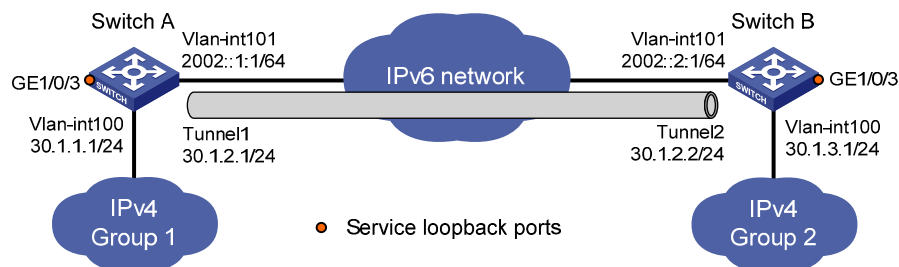
If you specify a source interface instead of a source address for the tunnel, the source address of the tunnel is the primary IP address of the source interface.

Configuration example

Network requirements

As shown in Figure 86, the two subnets Group 1 and Group 2 in the private network running IPv4 are interconnected over the IPv6 network by using an IPv4 over IPv6 tunnel between Switch A and Switch B.

Figure 86 Network diagram for an IPv4 over IPv6 tunnel



Configuration procedure

Make sure that Switch A and Switch B have the corresponding VLAN interfaces created and are reachable to each other.

- Configuration on Switch A

Enable IPv6.

```
<SwitchA> system-view
```

```
[SwitchA] ipv6
```



```

# Configure an IPv4 address for VLAN-interface 100.
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ip address 30.1.1.1 255.255.255.0
[SwitchA-Vlan-interface100] quit

# Configure an IPv6 address for VLAN-interface 101
[SwitchA] interface vlan-interface 101
[SwitchA-Vlan-interface101] ipv6 address 2002::1:1 64
[SwitchA-Vlan-interface101] quit

# Create the interface tunnel 1.
[SwitchA] interface tunnel 1

# Configure an IPv4 address for the interface tunnel 1.
[SwitchA-Tunnel1] ip address 30.1.2.1 255.255.255.0

# Configure the tunnel encapsulation mode.
[SwitchA-Tunnel1] tunnel-protocol ipv4-ipv6

# Configure the source address for the interface Tunnel 1 (IP address of VLAN-interface 101).
[SwitchA-Tunnel1] source 2002::1:1

# Configure the destination address of the interface Tunnel 1 (IP address of VLAN-interface 101 of Switch B).
[SwitchA-Tunnel1] destination 2002::2:1
[SwitchA-Tunnel1] quit

# Create service loopback group 1 to support the tunnel service.
[SwitchA] service-loopback group 1 type tunnel

# Add GigabitEthernet1/0/3 to service loopback group 1.
[SwitchA] interface GigabitEthernet 1/0/3
[SwitchA-GigabitEthernet1/0/3] undo stp enable
[SwitchA-GigabitEthernet1/0/3] undo ndp enable
[SwitchA-GigabitEthernet1/0/3] undo lldp enable
[SwitchA-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchA-GigabitEthernet1/0/3] quit

# Reference service loopback group 1 on the tunnel.
[SwitchA] interface tunnel 1
[SwitchA-Tunnel1] service-loopback-group 1
[SwitchA-Tunnel1] quit

# Configure a static route from Switch A through the interface Tunnel 1 to Group 2.
[SwitchA] ip route-static 30.1.3.0 255.255.255.0 tunnel 1

```

- Configuration on Switch B

```

# Enable IPv6.
<SwitchA> system-view
[SwitchA] ipv6

```

```

# Configure an IPv4 address for VLAN-interface 100.
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ip address 30.1.3.1 255.255.255.0
[SwitchB-Vlan-interface100] quit

# Configure an IPv6 address for VLAN-interface 101.
[SwitchB] interface vlan-interface 101
[SwitchB-Vlan-interface101] ipv6 address 2002::2:1 64
[SwitchB-Vlan-interface101] quit

# Create the interface Tunnel 2.
[SwitchB] interface tunnel 2

# Configure an IPv4 address for the interface Tunnel 2.
[SwitchB-Tunnel2] ip address 30.1.2.2 255.255.255.0

# Configure the tunnel encapsulation mode.
[SwitchB-Tunnel2] tunnel-protocol ipv4-ipv6

# Configure the source address for the interface Tunnel 2 (IP address of VLAN-interface 101).
[SwitchB-Tunnel2] source 2002::2:1

# Configure the destination address for the interface Tunnel 2 (IP address of VLAN-interface 101 of Switch A).
[SwitchB-Tunnel2] destination 2002::1:1
[SwitchB-Tunnel2] quit

# Create service loopback group 1 to support the tunnel service.
[SwitchB] service-loopback group 1 type tunnel

# Add GigabitEthernet1/0/3 to service loopback group 1.
[SwitchB] interface GigabitEthernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] undo stp enable
[SwitchB-GigabitEthernet1/0/3] undo ndp enable
[SwitchB-GigabitEthernet1/0/3] undo lldp enable
[SwitchB-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchB-GigabitEthernet1/0/3] quit

# Reference service loopback group 1 on the tunnel.
[SwitchB] interface tunnel 2
[SwitchB-Tunnel2] service-loopback-group 1
[SwitchB-Tunnel2] quit

# Configure a static route from Switch B through the interface Tunnel 2 to Group 1.
[SwitchB] ip route-static 30.1.1.0 255.255.255.0 tunnel 2

```

Configuration verification

After the configuration, display the status of the tunnel interfaces on Switch A and Switch B, respectively.

```
[SwitchA] display interface tunnel 1
Tunnell current state: UP
Line protocol current state: UP
Description: Tunnell Interface
The Maximum Transmit Unit is 1460
Internet Address is 30.1.2.1/24 Primary
Encapsulation is TUNNEL, service-loopback-group ID is 1.
Tunnel source 2002::0001:0001, destination 2002::0002:0001
Tunnel protocol/transport IP/IPv6
    Last 300 seconds input:  0 bytes/sec, 0 packets/sec
    Last 300 seconds output: 0 bytes/sec, 0 packets/sec
    152 packets input,  9728 bytes
    0 input error
    168 packets output, 10752 bytes
    0 output error
```

```
[SwitchB] display interface tunnel 2
Tunnel2 current state: UP
Line protocol current state: UP
Description: Tunnel2 Interface
The Maximum Transmit Unit is 1460
Internet Address is 30.1.2.2/24 Primary
Encapsulation is TUNNEL, service-loopback-group ID is 1.
Tunnel source 2002::0002:0001, destination 2002::0001:0001
Tunnel protocol/transport IP/IPv6
    Last 300 seconds input:  1 bytes/sec, 0 packets/sec
    Last 300 seconds output: 1 bytes/sec, 0 packets/sec
    167 packets input, 10688 bytes
    0 input error
    170 packets output, 10880 bytes
    0 output error
```

Ping the IPv4 address of the peer interface VLAN-interface 100 from Switch A.

```
[RouterA] ping 30.1.3.1
PING 30.1.3.1: 56 data bytes, press CTRL_C to break
    Reply from 30.1.3.1: bytes=56 Sequence=1 ttl=255 time=46 ms
    Reply from 30.1.3.1: bytes=56 Sequence=2 ttl=255 time=15 ms
    Reply from 30.1.3.1: bytes=56 Sequence=3 ttl=255 time=16 ms
    Reply from 30.1.3.1: bytes=56 Sequence=4 ttl=255 time=15 ms
    Reply from 30.1.3.1: bytes=56 Sequence=5 ttl=255 time=16 ms

--- 30.1.3.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 15/21/46 ms
```

Configuring an IPv6 over IPv6 tunnel

Configuration prerequisites

Configure IP addresses for interfaces (such as the VLAN interface, and loopback interface) on the device to ensure normal communication. One of the interfaces is used as the source interface of the tunnel.

Configuration procedure

| Step | Command | Remarks | |
|--|--|---|---|
| 1. Enter system view. | system-view | — | |
| 2. Enable IPv6. | ipv6 | Required. By default, the IPv6 packet forwarding function is disabled. | |
| 3. Enter tunnel interface view. | Interface tunnel number | — | |
| 4. Configure an IPv6 address for the tunnel interface. | Configure an IPv6 global unicast address or site-local address. | ipv6 address { ipv6-address prefix-length ipv6-address/prefix-length } | Required. |
| | Configure an IPv6 link-local address. | ipv6 address ipv6-address/prefix-length eui-64 | Use one of the commands. By default, no IPv6 address is configured for the tunnel interface. |
| | | ipv6 address auto link-local ipv6 address ipv6-address link-local | |
| 5. Specify the IPv6 over IPv6 tunnel mode. | tunnel-protocol ipv6-ipv6 | Optional. By default, the tunnel is a GRE over IPv4 tunnel. The same tunnel mode should be configured at both ends of the tunnel. Otherwise, packet delivery fails. | |
| 6. Configure a source address or interface for the tunnel interface. | source { ipv6-address interface-type interface-number } | Required. By default, no source address or interface is configured for the tunnel. | |
| 7. Configure the destination address for the tunnel interface. | destination ipv6-address | Required. By default, no destination address is configured for the tunnel. | |

To encapsulate and forward IPv6 packets whose destination address does not belong to the subnet where the receiving tunnel interface resides, configure a static route or dynamic routing for forwarding those packets through this tunnel interface. If you configure a static route to that destination IPv6 address, specify this tunnel interface as the outbound interface, or the peer tunnel interface address as the next hop. A similar configuration needs to be performed at the other tunnel end. If you configure dynamic routing at both ends, enable the dynamic routing protocol on both tunnel interfaces. For the detailed configuration, see *Layer 3—IP Routing Configuration Guide*.

The IPv6 address of a tunnel interface must not be on the same subnet as the destination address of the tunnel.

The destination address of a route with the tunnel interface as the egress interface must not be on the same subnet as the destination address of the tunnel.

Two or more tunnel interfaces using the same encapsulation protocol must have different source and destination addresses.

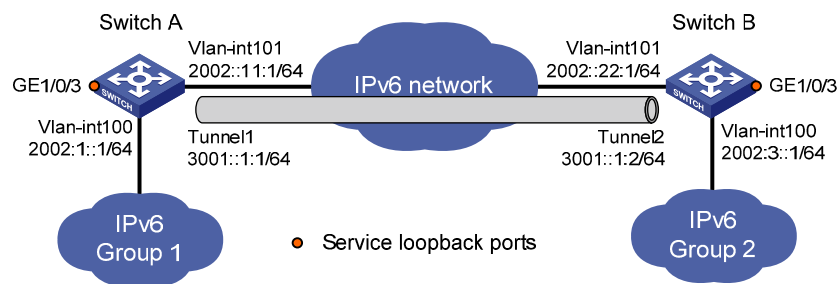
If you specify a source interface instead of a source address for the tunnel, the source address of the tunnel is the primary IP address of the source interface.

Configuration example

Network requirements

As shown in [Figure 87](#), the two subnets Group 1 and Group 2 running IPv6 are interconnected by using an IPv6 over IPv6 tunnel between Switch A and Switch B.

Figure 87 Network diagram for an IPv6 over IPv6 tunnel



Configuration procedure

Make sure that Switch A and Switch B have the corresponding VLAN interfaces created and are reachable to each other.

- Configuration on Switch A

Enable IPv6.

```
<SwitchA> system-view
[SwitchA] ipv6
```

Configure an IPv6 address for VLAN-interface 100.

```
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ipv6 address 2002:1::1 64
[SwitchA-Vlan-interface100] quit
```

Configure an IPv6 address for VLAN-interface 101.

```
[SwitchA] interface vlan-interface 101
[SwitchA-Vlan-interface101] ipv6 address 2002::11:1 64
[SwitchA-Vlan-interface101] quit
```

```

# Create the interface tunnel 1.
[SwitchA] interface tunnel 1

# Configure an IPv6 address for the interface tunnel 1.
[SwitchA-Tunnel1] ipv6 address 3001::1:1 64

# Configure the tunnel encapsulation mode.
[SwitchA-Tunnel1] tunnel-protocol ipv6-ipv6

# Configure the source address for the interface Tunnel 1 (IP address of VLAN-interface 101).
[SwitchA-Tunnel1] source 2002:11::1

# Configure the destination address for the interface Tunnel 1 (IP address of VLAN-interface 101 of Switch B).
[SwitchA-Tunnel1] destination 2002::22:1
[SwitchA-Tunnel1] quit

# Create service loopback group 1 to support the tunnel service.
[SwitchA] service-loopback group 1 type tunnel

# Add GigabitEthernet1/0/3 to service loopback group 1.
[SwitchA] interface GigabitEthernet 1/0/3
[SwitchA-GigabitEthernet1/0/3] undo stp enable
[SwitchA-GigabitEthernet1/0/3] undo ndp enable
[SwitchA-GigabitEthernet1/0/3] undo lldp enable
[SwitchA-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchA-GigabitEthernet1/0/3] quit

# Reference service loopback group 1 on the tunnel.
[SwitchA] interface tunnel 1
[SwitchA-Tunnel1] service-loopback-group 1
[SwitchA-Tunnel1] quit

# Configure a static route from Switch A through the interface Tunnel 1 to Group 2.
[SwitchA] ipv6 route-static 2002:3:: 64 tunnel 1

```

- Configuration on Switch B

```

# Enable IPv6.
<SwitchB> system-view
[SwitchB] ipv6

# Configure an IPv6 address for VLAN-interface 100.
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ipv6 address 2002:3::1 64
[SwitchB-Vlan-interface100] quit

# Configure an IPv6 address for VLAN-interface 101.
[SwitchB] interface vlan-interface 101
[SwitchB-Vlan-interface101] ipv6 address 2002::22:1 64
[SwitchB-Vlan-interface101] quit

# Create the interface Tunnel 2.
[SwitchB] interface tunnel 2

```

```

# Configure an IPv6 address for the interface Tunnel 2.
[SwitchB-Tunnel2] ipv6 address 3001::1:2 64

# Configure the tunnel encapsulation mode.
[SwitchB-Tunnel2] tunnel-protocol ipv6-ipv6

# Configure the source address for the interface Tunnel 2 (IP address of VLAN-interface 101)
[SwitchB-Tunnel2] source 2002::22:1

# Configure the destination address for the interface Tunnel 2 (IP address of VLAN-interface 101 of Switch A).
[SwitchB-Tunnel2] destination 2002::11:1
[SwitchB-Tunnel2] quit

# Create service loopback group 1 to support the tunnel service.
[SwitchB] service-loopback group 1 type tunnel

# Add GigabitEthernet1/0/3 to service loopback group 1.
[SwitchB] interface GigabitEthernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] undo stp enable
[SwitchB-GigabitEthernet1/0/3] undo ndp enable
[SwitchB-GigabitEthernet1/0/3] undo lldp enable
[SwitchB-GigabitEthernet1/0/3] port service-loopback group 1
[SwitchB-GigabitEthernet1/0/3] quit

# Reference service loopback group 1 on the tunnel.
[SwitchB] interface tunnel 2
[SwitchB-Tunnel2] service-loopback-group 1
[SwitchB-Tunnel2] quit

# Configure a static route from Switch B through the interface Tunnel 2 to Group 1.
[SwitchB] ipv6 route-static 2002:1:: 64 tunnel 2

```

Configuration verification

After the above configuration, display the status of the tunnel interfaces on Switch A and Switch B, respectively.

```

[SwitchA] display ipv6 interface tunnel 1 verbose
Tunnell1 current state :UP
Line protocol current state :UP
IPv6 is enabled, link-local address is FE80::2013:1
  Global unicast address(es):
    3001::1:1, subnet is 3001::/64
  Joined group address(es):
    FF02::1:FF13:1
    FF02::1:FF01:1
    FF02::1:FF00:0
    FF02::2
    FF02::1
MTU is 1460 bytes
ND reachable time is 30000 milliseconds
ND retransmit interval is 1000 milliseconds
Hosts use stateless autoconfig for addresses

```

```

IPv6 Packet statistics:
...
[SwitchB] display ipv6 interface tunnel 2 verbose
Tunnel2 current state :UP
Line protocol current state :UP
IPv6 is enabled, link-local address is FE80::2024:1
  Global unicast address(es):
    3001::1:2, subnet is 3001::/64
  Joined group address(es):
    FF02::1:FF24:1
    FF02::1:FF01:2
    FF02::1:FF00:0
    FF02::2
    FF02::1
  MTU is 1460 bytes
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
...
# Ping the IPv6 address of the peer interface VLAN-interface 100 from Switch A.
[SwitchA] ping ipv6 2002:3::1
  PING 2002:3::1 : 56 data bytes, press CTRL_C to break
  Reply from 2002:3::1
    bytes=56 Sequence=1 hop limit=64 time = 31 ms
  Reply from 2002:3::1
    bytes=56 Sequence=2 hop limit=64 time = 1 ms
  Reply from 2002:3::1
    bytes=56 Sequence=3 hop limit=64 time = 16 ms
  Reply from 2002:3::1
    bytes=56 Sequence=4 hop limit=64 time = 16 ms
  Reply from 2002:3::1
    bytes=56 Sequence=5 hop limit=64 time = 31 ms

  --- 2002:3::1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
  round-trip min/avg/max = 1/19/31 ms

```

Configuring a GRE over IPv4 tunnel

Configuration prerequisites

Interfaces on a device, such as VLAN interfaces and loopback interfaces, are configured with IPv4 addresses and can communicate. These interfaces can be used as the source of a virtual tunnel interface to ensure the reachability of the tunnel destination address.

Configuration procedure

| Step | Command | Remarks |
|--|--|--|
| 1. Enter system view. | system-view | — |
| 2. Create a tunnel interface and enter tunnel interface view. | interface tunnel <i>interface-number</i> | Required. By default, a device has no tunnel interface. |
| 3. Configure an IPv4 address for the tunnel interface. | ip address <i>ip-address</i> { <i>mask</i> <i>mask-length</i> } | Required. By default, a tunnel interface has no IPv4 address. |
| 4. Set the tunnel mode to GRE over IPv4. | tunnel-protocol gre | Optional. By default, the tunnel is a GRE over IPv4 tunnel You must configure the same tunnel mode on both ends of a tunnel. Otherwise, packet delivery fails. |
| 5. Configure the source address or interface for the tunnel interface. | source { <i>ip-address</i> <i>interface-type interface-number</i> } | Required. By default, no source address or interface is configured for a tunnel interface. |
| 6. Configure the destination address for the tunnel interface. | destination <i>ip-address</i> | Required. By default, no destination address is configured for a tunnel interface. |
| 7. Configure a route through the tunnel. | See the routing protocols in Layer 3 – IP Routing Configuration Guide. | Optional. Each end of the tunnel must have a route (static or dynamic) through the tunnel to the other end. |

The source address and destination address of a tunnel uniquely identify a path. They must be configured at both ends of the tunnel and the source address at one end must be the destination address at the other end and vice versa.

Tunnel interfaces using the same encapsulation protocol must have different source addresses and destination addresses.

If you configure a source interface for a tunnel interface, the tunnel interface takes the primary IP address of the source interface as its source address.

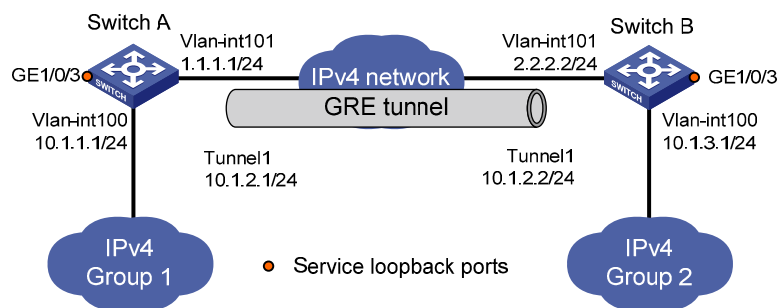
When configuring a route through the tunnel, configure a static route, using the address of the network segment that the original packet is destined for as its destination address and the address of the peer tunnel interface as its next hop. Alternately, enable a dynamic routing protocol on both the tunnel interface and the router interface connecting the private network.

Configuration example

Network requirements

Switch A and Switch B are interconnected through the Internet. Two private IPv4 subnets Group 1 and Group 2 are interconnected through a GRE tunnel between the two switches.

Figure 88 Network diagram for a GRE over IPv4 tunnel



Configuration procedure

Before the configuration, make sure that Switch A and Switch B are reachable to each other.

- Configure Switch A

Configure an IPv4 address .

```
<SwitchA> system-view
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ip address 10.1.1.1 255.255.255.0
[SwitchA-Vlan-interface100] quit
[SwitchA] interface vlan-interface 101
[SwitchA-Vlan-interface101] ip address 1.1.1.1 255.255.255.0
[SwitchA-Vlan-interface101] quit
```

Create an interface named **Tunnel 1**.

```
[SwitchA] interface tunnel 1
```

Configure an IPv4 address for interface Tunnel 1.

```
[SwitchA-Tunnel1] ip address 10.1.2.1 255.255.255.0
```

Configure the tunnel encapsulation mode.

```
[SwitchA-Tunnel1] tunnel-protocol gre
```

Configure the source address of interface Tunnel 1 to be the IP address of the VLAN interface.

```
[SwitchA-Tunnel1] source vlan-interface 101
```

```

# Configure the destination address for interface Tunnel 1.
[SwitchA-Tunnel1] destination 2.2.2.2
[SwitchA-Tunnel1] quit

# Create service loopback group 1, setting the service type to tunnel.
[SwitchA] service-loopback group 1 type tunnel

# Add interface GigabitEthernet 1/0/3 to service loopback group 1.
[SwitchA] interface GigabitEthernet 1/0/3
[SwitchA-GigabitEthernet1/0/3] undo stp enable
[SwitchA-GigabitEthernet1/0/3] undo ndp enable
[SwitchA-GigabitEthernet1/0/3] undo lldp enable
[SwitchA-GigabitEthernet1/0/3] port service-loopback group 1

# Apply service loopback group 1 to the tunnel in tunnel interface view.
[SwitchA-GigabitEthernet1/0/3] quit
[SwitchA] interface tunnel 1
[SwitchA-Tunnel1] service-loopback-group 1
[SwitchA-Tunnel1] quit

# Configure a static route from Switch A through interface Tunnel 1 to Group 2.
[SwitchA] ip route-static 10.1.3.0 255.255.255.0 tunnel 1

```

- **Configure Switch B**

```

# Configure an IPv4 address.
<SwitchB> system-view
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ip address 10.1.3.1 255.255.255.0
[SwitchB-Vlan-interface100] quit
[SwitchB] interface vlan-interface 101
[SwitchB-Vlan-interface101] ip address 2.2.2.2 255.255.255.0
[SwitchB-Vlan-interface101] quit

# Create an interface named Tunnel 1.
[SwitchB] interface tunnel 1

# Configure an IPv4 address for interface Tunnel 1.
[SwitchB-Tunnel1] ip address 10.1.2.2 255.255.255.0

# Configure the tunnel encapsulation mode.
[SwitchB-Tunnel1] tunnel-protocol gre

# Configure the source address for interface Tunnel 1.
[SwitchB-Tunnel1] source vlan-interface 101

# Configure the destination address for interface Tunnel 1.
[SwitchB-Tunnel1] destination 1.1.1.1
[SwitchB-Tunnel1] quit

# Create service loopback group 1, setting the service type to tunnel.
[SwitchB] service-loopback group 1 type tunnel

```

```

# Add interfaceGigabitEthernet 1/0/3 to service loopback group 1.
[SwitchB] interface GigabitEthernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] undo stp enable
[SwitchB-GigabitEthernet1/0/3] undo ndp enable
[SwitchB-GigabitEthernet1/0/3] undo lldp enable
[SwitchB-GigabitEthernet1/0/3] port service-loopback group 1

# Apply service loopback group 1 to the tunnel in tunnel interface view.
[SwitchB-GigabitEthernet1/0/3] quit
[SwitchB] interface tunnel 1
[SwitchB-Tunnel1] service-loopback-group 1
[SwitchB-Tunnel1] quit

# Configure a static route from Switch B through interface Tunnel 1 to Group 1.
[SwitchB] ip route-static 10.1.1.0 255.255.255.0 Tunnel 1

```

Configuration verification

After the configuration, view the tunnel interface status on Switch A and Switch B respectively.

```

[SwitchA] display interface tunnel 1
Tunnel1 current state: UP
Line protocol current state: UP
Description: Tunnel1 Interface
The Maximum Transmit Unit is 1476
Internet Address is 10.1.2.1/24 Primary
Encapsulation is TUNNEL, service-loopback-group ID is 1.
Tunnel source 1.1.1.1 (Vlan-interface101), destination 2.2.2.2
Tunnel bandwidth 64 (kbps)
Tunnel protocol/transport GRE/IP
    GRE key disabled
    Checksumming of GRE packets disabled
Last clearing of counters: Never
    Last 300 seconds input:  0 bytes/sec, 0 packets/sec
    Last 300 seconds output: 0 bytes/sec, 0 packets/sec
    10 packets input,  840 bytes
    0 input error
    10 packets output,  840 bytes
    0 output error

[SwitchB] display interface tunnel 1
Tunnel1 current state: UP
Line protocol current state: UP
Description: Tunnel1 Interface
The Maximum Transmit Unit is 1476
Internet Address is 10.1.2.2/24 Primary
Encapsulation is TUNNEL, service-loopback-group ID is 1.
Tunnel source 2.2.2.2 (Vlan-interface101), destination 1.1.1.1
Tunnel bandwidth 64 (kbps)
Tunnel protocol/transport GRE/IP
    GRE key disabled
    Checksumming of GRE packets disabled

```

```

Last clearing of counters: Never
  Last 300 seconds input:  2 bytes/sec, 0 packets/sec
  Last 300 seconds output: 2 bytes/sec, 0 packets/sec
  10 packets input,  840 bytes
  0 input error
  10 packets output, 840 bytes
  0 output error

# From Switch B, ping the IP address of VLAN-interface 100 on Switch A.
[SwitchB] ping 10.1.1.1
  PING 10.1.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=255 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=255 time=2 ms

--- 10.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/2/2 ms

```

Configuring a GRE over IPv6 tunnel

Configuration prerequisites

Interfaces on a device, such as VLAN interfaces and loopback interfaces, are configured with IPv6 addresses and can communicate. These interfaces can serve as the source of a virtual tunnel interface to ensure the reachability of the destination address.

Configuration procedure

| Step | Command | Remarks |
|---|---|--|
| 1. Enter system view. | system-view | — |
| 2. Enable the IPv6 packet forwarding function. | ipv6 | Required. Disabled by default |
| 3. Create a tunnel interface and enter tunnel interface view. | interface tunnel <i>interface-number</i> | Required. By default, there is no tunnel interface on a device. |
| 4. Configure an IPv4 address for the tunnel interface. | ip address ip-address { mask mask-length } | Required. By default, no IPv4 address is configured for a tunnel interface. |

| Step | Command | Remarks |
|--|--|--|
| 5. Set the tunnel mode to GRE over IPv6. | tunnel-protocol gre ipv6 | Required. By default, the tunnel is a GRE over IPv4 tunnel You must configure the same tunnel mode on both ends of a tunnel. Otherwise, packet delivery fails. |
| 6. Configure the source address or interface for the tunnel interface. | source { ipv6-address interface-type interface-number } | Required. By default, no source address or interface is configured for a tunnel interface. |
| 7. Configure the destination address for the tunnel interface. | destination ipv6-address | Required. By default, no destination address is configured for a tunnel interface. |
| 8. Configure a route through the tunnel. | See the routing protocols in Layer 3 – IP Routing Configuration Guide. | Optional. Each end of the tunnel must have a route (static or dynamic) through the tunnel to the other end. |

If you delete a tunnel interface, the functions configured on this tunnel interface is removed as well.

The source address and destination address of a tunnel uniquely identify a path. They must be configured at both ends of the tunnel and the source address at one end must be the destination address at the other end and vice versa.

Tunnel interfaces using the same encapsulation protocol must have different source addresses and destination addresses.

If you configure a source interface for a tunnel interface, the tunnel interface takes the primary IP address of the source interface as its source address.

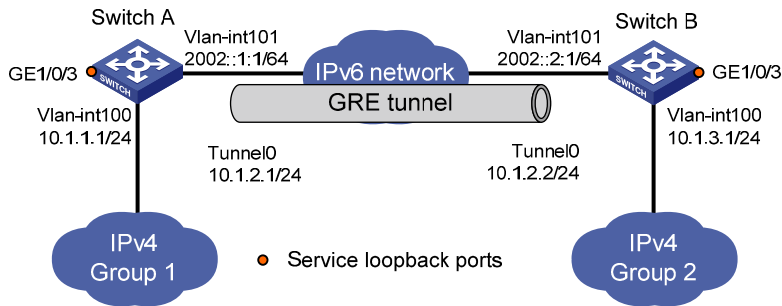
When configuring a route through the tunnel, configure a static route, using the address of the network segment the original packet is destined for as its destination address and the address of the peer tunnel interface as its next hop. Or, enable a dynamic routing protocol on both the tunnel interface and the router interface connecting the private network, so that the dynamic routing protocol can establish a routing entry that allows the tunnel to forward packets through the tunnel. It is not allowed to set up a static route whose destination address is in the subnet of the tunnel interface.

Configuration example

Network requirements

Two IPv4 subnets Group 1 and Group 2 are interconnected through a GRE tunnel over the IPv6 network between Switch A and Switch B.

Figure 89 Network diagram for a GRE over IPv6 tunnel



Configuration procedure

Before the configuration, make sure that Switch A and Switch B are reachable to each other.

- Configure Switch A

```
<SwitchA> system-view
```

```
# Enable IPv6.
```

```
[SwitchA] ipv6
```

```
# Configure interface VLAN-interface 100.
```

```
[SwitchA] interface vlan-interface 100
```

```
[SwitchA-Vlan-interface100] ip address 10.1.1.1 255.255.255.0
```

```
[SwitchA-Vlan-interface100] quit
```

```
# Configure interface VLAN-interface 101, the physical interface of the tunnel.
```

```
[SwitchA] interface vlan-interface 101
```

```
[SwitchA-Vlan-interface101] ipv6 address 2002::1 64
```

```
[SwitchA-Vlan-interface101] quit
```

```
# Create an interface named Tunnel 0.
```

```
[SwitchA] interface tunnel 0
```

```
# Configure an IPv4 address for interface Tunnel 0.
```

```
[SwitchA-Tunnel0] ip address 10.1.2.1 255.255.255.0
```

```
# Configure the tunnel encapsulation mode.
```

```
[SwitchA-Tunnel0] tunnel-protocol gre ipv6
```

```
# Configure the source address of interface Tunnel 0 to be the IP address of interface VLAN-interface 100.
```

```
[SwitchA-Tunnel0] source 2002::1:1
```

```
# Configure the destination address of interface Tunnel 0 to be the IP address of interface VLAN-interface 101 on Switch B.
```

```
[SwitchA-Tunnel0] destination 2002::2:1
```

```
[SwitchA-Tunnel0] quit
```

```
# Create service loopback group 1, setting the service type to tunnel.
```

```
[SwitchA] service-loopback group 1 type tunnel
```

```

# Add interface GigabitEthernet 1/0/3 to service loopback group 1.
[SwitchA] interface GigabitEthernet 1/0/3
[SwitchA-GigabitEthernet1/0/3] undo stp enable
[SwitchA-GigabitEthernet1/0/3] undo ndp enable
[SwitchA-GigabitEthernet1/0/3] undo lldp enable
[SwitchA-GigabitEthernet1/0/3] port service-loopback group 1

# Apply service loopback group 1 to the tunnel in tunnel interface view.
[SwitchA-GigabitEthernet1/0/3] quit
[SwitchA] interface tunnel 0
[SwitchA-Tunnel0] service-loopback-group 1
[SwitchA-Tunnel0] quit

# Configure a static route from Switch A through interface Tunnel 0 to Group 2.
[SwitchA] ip route-static 10.1.3.0 255.255.255.0 tunnel 0

```

- **Configure Switch B**

```

<SwitchB> system-view

# Enable IPv6.
[SwitchB] ipv6

# Configure interface VLAN-interface 100.
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ip address 10.1.3.1 255.255.255.0
[SwitchB-Vlan-interface100] quit

# Configure interface VLAN-interface 101, the physical interface of the tunnel.
[SwitchB] interface vlan-interface 101
[SwitchB-Vlan-interface101] ipv6 address 2002::2:1 64
[SwitchB-Vlan-interface101] quit

# Create an interface named Tunnel 0.
[SwitchB] interface tunnel 0

# Configure an IPv4 address for interface Tunnel 0.
[SwitchB-Tunnel0] ip address 10.1.2.2 255.255.255.0

# Configure the tunnel encapsulation mode.
[SwitchB-Tunnel0] tunnel-protocol gre ipv6

# Configure the source address of interface Tunnel 0 to be the IP address of interface VLAN-interface 101.
[SwitchB-Tunnel0] source 2002::2:1

# Configure the destination address of interface Tunnel 0 to be the IP address of interface VLAN-interface 101 on Switch A.
[SwitchB-Tunnel0] destination 2002::1:1
[SwitchB-Tunnel0] quit

# Create service loopback group 1, setting the service type to tunnel.
[SwitchB] service-loopback group 1 type tunnel

```



```

# Add interface GigabitEthernet 1/0/3 to service loopback group 1.
[SwitchB] interface GigabitEthernet 1/0/3
[SwitchB-GigabitEthernet1/0/3] undo stp enable
[SwitchB-GigabitEthernet1/0/3] undo ndp enable
[SwitchB-GigabitEthernet1/0/3] undo lldp enable
[SwitchB-GigabitEthernet1/0/3] port service-loopback group 1

# Apply service loopback group 1 to the tunnel in tunnel interface view.
[SwitchB-GigabitEthernet1/0/3] quit
[SwitchB] interface tunnel 0
[SwitchB-Tunnel0] service-loopback-group 1
[SwitchB-Tunnel0] quit

# Configure a static route from Switch B through interface Tunnel 0 to Group 1.
[SwitchB] ip route-static 10.1.1.0 255.255.255.0 tunnel 0

```

Configuration verification

After the configuration, view the tunnel interface status on Switch A and Switch B respectively.

```

[SwitchA] display interface Tunnel 0
Tunnel0 current state: UP
Line protocol current state: UP
Description: Tunnel0 Interface
The Maximum Transmit Unit is 1456
Internet Address is 10.1.2.1/24 Primary
Encapsulation is TUNNEL, service-loopback-group ID is 1.
Tunnel source 2002::1:1, destination 2002::2:1
Tunnel bandwidth 64 (kbps)
Tunnel protocol/transport GRE/IPV6
    GRE key disabled
    Checksumming of GRE packets disabled
Last clearing of counters: Never
    Last 300 seconds input:  0 bytes/sec, 0 packets/sec
    Last 300 seconds output: 0 bytes/sec, 0 packets/sec
    10 packets input,  840 bytes
    0 input error
    10 packets output, 840 bytes
    0 output error

[SwitchB] display interface Tunnel 0
Tunnel0 current state: UP
Line protocol current state: UP
Description: Tunnel0 Interface
The Maximum Transmit Unit is 1456
Internet Address is 10.1.2.2/24 Primary
Encapsulation is TUNNEL, service-loopback-group ID is 1.
Tunnel source 2002::2:1, destination 2002::1:1
Tunnel bandwidth 64 (kbps)
Tunnel protocol/transport GRE/IPV6
    GRE key disabled
    Checksumming of GRE packets disabled

```

```

Last clearing of counters: Never
  Last 300 seconds input:  0 bytes/sec, 0 packets/sec
  Last 300 seconds output: 0 bytes/sec, 0 packets/sec
  10 packets input,  840 bytes
  0 input error
  10 packets output, 840 bytes
  0 output error

```

From Switch B, ping the IP address of VLAN-interface 100 on Switch A.

```

[SwitchB] ping 10.1.1.1
  PING 10.1.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=255 time=3 ms
  Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=255 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=255 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=255 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=255 time=3 ms

--- 10.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/2/3 ms

```

Displaying and maintaining tunneling configuration

| Task | Command | Remarks |
|--|---|------------------------|
| Display information about tunnel interfaces. | display interface [tunnel] [brief [down]] [{ begin exclude include } <i>regular-expression</i>] display interface tunnel <i>number</i> [brief] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Display IPv6 information on tunnel interfaces. | display ipv6 interface tunnel [<i>number</i>] [verbose] [{ begin exclude include } <i>regular-expression</i>] | Available in any view |
| Clear statistics on tunnel interfaces. | reset counters interface [tunnel [<i>number</i>]] | Available in user view |

Troubleshooting tunneling configuration

Symptom

After the configuration of related parameters such as tunnel source address, tunnel destination address, and tunnel mode, the tunnel interface is still not up.

Solution

Follow these steps:

1. The common cause is that the physical interface of the tunnel source is not up. Use **display interface tunnel** or **display ipv6 interface tunnel** to view whether the physical interface of the tunnel source is up. If the physical interface is down, use **debugging tunnel event** in user view to view the cause.
2. Another possible cause is that the tunnel destination is unreachable. Use **display ipv6 routing-table** or **display ip routing-table** to view whether the tunnel destination is reachable. If no routing entry is available for tunnel communication in the routing table, configure related routes.

Support and other resources

Contacting HP

For worldwide technical support information, see the HP support website:

<http://www.hp.com/support>

Before contacting HP, collect the following information:

- Product model names and numbers
- Technical support registration number (if applicable)
- Product serial numbers
- Error messages
- Operating system type and revision level
- Detailed questions

Subscription service

HP recommends that you register your product at the Subscriber's Choice for Business website:

<http://www.hp.com/go/wwalerts>

After registering, you will receive email notification of product enhancements, new driver versions, firmware updates, and other product resources.

Related information

Documents

To find related documents, browse to the Manuals page of the HP Business Support Center website:

<http://www.hp.com/support/manuals>

- For related documentation, navigate to the Networking section, and select a networking category.
- For a complete list of acronyms and their definitions, see *HP A-Series Acronyms*.

Websites

- HP.com <http://www.hp.com>
- HP Networking <http://www.hp.com/go/networking>
- HP manuals <http://www.hp.com/support/manuals>
- HP download drivers and software <http://www.hp.com/support/downloads>
- HP software depot <http://www.software.hp.com>

Conventions

This section describes the conventions used in this documentation set.





Command conventions

| Convention | Description |
|-------------------|--|
| Boldface | Bold text represents commands and keywords that you enter literally as shown. |
| <i>Italic</i> | <i>Italic</i> text represents arguments that you replace with actual values. |
| [] | Square brackets enclose syntax choices (keywords or arguments) that are optional. |
| { x y ... } | Braces enclose a set of required syntax choices separated by vertical bars, from which you select one. |
| [x y ...] | Square brackets enclose a set of optional syntax choices separated by vertical bars, from which you select one or none. |
| { x y ... } * | Asterisk-marked braces enclose a set of required syntax choices separated by vertical bars, from which you select at least one. |
| [x y ...] * | Asterisk-marked square brackets enclose optional syntax choices separated by vertical bars, from which you select one choice, multiple choices, or none. |
| &<1-n> | The argument or keyword and argument combination before the ampersand (&) sign can be entered 1 to n times. |
| # | A line that starts with a pound (#) sign is comments. |

GUI conventions

| Convention | Description |
|-----------------|--|
| Boldface | Window names, button names, field names, and menu items are in bold text. For example, the New User window appears; click OK . |
| > | Multi-level menus are separated by angle brackets. For example, File > Create > Folder . |

Symbols

| Convention | Description |
|--|--|
|  WARNING | An alert that calls attention to important information that if not understood or followed can result in personal injury. |
|  CAUTION | An alert that calls attention to important information that if not understood or followed can result in data loss, data corruption, or damage to hardware or software. |
|  IMPORTANT | An alert that calls attention to essential information. |
| NOTE | An alert that contains additional or supplementary information. |
|  TIP | An alert that provides helpful information. |

Network topology icons



Represents a generic network device, such as a router, switch, or firewall.



Represents a routing-capable device, such as a router or Layer 3 switch.



Represents a generic switch, such as a Layer 2 or Layer 3 switch, or a router that supports Layer 2 forwarding and other Layer 2 features.

Port numbering in examples

The port numbers in this document are for illustration only and might be unavailable on your device.

Index

address

address/prefix assignment (DHCPv6), 149

address/prefix lease renewal (DHCPv6), 150

addressing. See IP addressing

allocation (DHCP), 27

anycast type (IPv6), 119

applying DHCPv6 address pool to an interface, 156

applying extended address pool on interface (DHCP), 47

assigning to interface (IP addressing), 23

autoconfiguration (IPv6 NDP), 121, 123

automatic configuration (IPv6), 117

configuring anycast address (IPv6), 130

configuring automatic generation of link-local address (IPv6), 129

configuring common address pool allocation mode (DHCP), 38

configuring dynamic address allocation (DHCP extended pool), 40

configuring dynamic address allocation (DHCP), 39

configuring global unicast (IPv6), 126

configuring interface to generate EUI-64 IPv6 address (IPv6), 126

configuring IP address conflict detection (DHCP), 48

configuring link-local address (IPv6), 128

configuring link-local address manually (IPv6), 129

configuring relay agent to release IP address (DHCP), 62

configuring server address pool (DHCP), 37

configuring stateless address autoconfiguration (IPv6), 127

configuring static allocation (DHCP), 38

creating address pool (DHCP), 38

creating pool (DHCPv6), 155

DHCP IP address lease extension, 28

DHCPv6 IA, 154

DHCPv6 multicast, 153

DHCPv6 PD, 154

DHCPv6 prefix selection process, 154

duplicate detection (IPv6 NDP), 122

ensuring client obtains authorized IP address from server (DHCP), 70

EUI-64 address-based interface identifier (IPv6), 120

format (IPv6), 118

four message assignment (DHCPv6), 150

hierarchical structure (IPv6), 117

IP address allocation sequence (DHCP), 36

IPv6, 118

lifetime (IP), 111

multicast (IPv6), 120

multicast type (IPv6), 119

pool (DHCP), 35

pool selection (DHCP), 36

pool structure (DHCP), 35

pool type (DHCP), 35

preference (IP), 111

proxy-advertised IP addresses, 111

RA destination address, 111

resolution. See address resolution

space (IPv6), 117

specifying interface EUI-64 IPv6 address manually (IPv6), 127

- static PMTU for specified address (IPv6), 138
- two message rapid assignment (DHCPv6), 149
- type (IPv6), 119
- unicast (IPv6), 119
- unicast type (IPv6), 119

address resolution

- ARP configuration, 1, 4, 7
- ARP process, 2
- ARP table, 3
- ARP table dynamic entry, 3
- ARP table dynamic entry, 4
- ARP table static entry, 3, 4
- DNS spoofing, 85
- IPv6, 122
- IPv6 NDP, 121
- multicast ARP configuration, 9

addressing. See IP addressing

advertising interval (IRDP), 111

aging time (ARP), 5

allocating

- address (DHCP), 27
- configuring common address pool mode (DHCP), 38
- dynamic address allocation mode (DHCP extended pool), 40
- dynamic address allocation mode (DHCP), 39
- dynamic IP address allocation process (DHCP), 28
- IP address sequence (DHCP), 36
- mechanism (DHCP), 27
- static address allocation mode (DHCP), 38

application

- BOOTP, 81
- environment (DHCP), 35
- relay agent application environment (DHCP), 55
- relay agent environment (DHCPv6), 161
- server environment (DHCPv6), 153

- trusted port application environment (DHCP), 71

applying

- DHCPv6 address pool to an interface, 156
- extended address pool on interface (DHCP), 47

ARP

- address resolution process, 2
- configuration, 1, 4, 7
- configuring max number dynamic interface entries, 4
- configuring multicast ARP for NLB, 6
- configuring quick update, 6
- configuring static entry, 4
- displaying, 7
- displaying snooping, 20
- dynamic entry, 3
- enabling dynamic entry check, 5
- enabling relay agent offline detection (DHCP), 61
- enabling snooping on VLAN, 20
- functions, 1
- gratuitious ARP. See gratuitous ARP
- gratuitous. See gratuitous ARP
- local proxy configuration, 14
- maintaining, 7
- maintaining snooping, 20
- message format, 1
- multicast configuration, 9
- setting dynamic entry aging time, 5
- snooping configuration, 20
- snooping operation, 20
- static entry, 3
- table, 3
- table static entry, 3

assigning

- address/prefix (DHCPv6), 149
- address/prefix lease renewal (DHCPv6), 150

- dynamic IP address assignment (DHCP), 51
- four message assignment (DHCPv6), 150
- IP address to interface, 23
- two message rapid assignment (DHCPv6), 149
- attack
 - enabling relay agent starvation attack protection (DHCP), 60
 - enabling snooping starvation attack protection (DHCP), 77
- autoconfiguration (IPv6 stateless address), 127
- backing up entries (DHCP snooping), 76
- BIMS (DHCP), 42
- bootfile (DHCP), 44
- BOOTP
 - application, 81
 - client configuration, 81, 82
 - configuring interface to dynamically obtain IP address, 82
 - displaying client configuration, 82
 - dynamically obtain IP address, 81
 - protocols and standards, 81
- broadcast
 - enabling forwarding of broadcasts to network (IP performance optimization), 103
 - enabling reception of broadcasts by network (IP performance optimization), 103
 - UDP Helper configuration, 115, 116
- buffer
 - configuring TCP send/receive buffer size, 105
- class (IP address), 21
- client
 - BOOTP configuration, 81, 82
 - configuring (IPv4 DNS), 86
 - configuring BIMS server information (DHCP), 42
 - configuring bootfile name (DHCP), 44
 - configuring DNS server (DHCP), 41
 - configuring domain name suffix (DHCP), 41
 - configuring gateway (DHCP), 43
 - configuring IPv6 DNS client, 95
 - configuring NetBIOS node type (DHCP), 41
 - configuring relay agent dynamic client entry periodic refresh (DHCP), 59
 - configuring relay agent support for Option 82 (DHCP), 62
 - configuring snooping support for Option 82 (DHCP), 75
 - configuring TFTP server (DHCP), 44
 - configuring voice service client Option 184 parameter (DHCP), 43
 - configuring WINS server (DHCP), 41
 - creating static binding (DHCP), 59
 - DHCP configuration, 67
 - DHCPv6 configuration, 165
 - DHCPv6 IA, 154
 - DNS proxy, 84
 - DUID (DHCPv6), 153
 - enabling on interface (DHCP), 67
 - enabling relay agent address check (DHCP), 59
 - ensuring client obtains address from authorized server (DHCPv6), 168
 - ensuring client obtains authorized IP address from server (DHCP), 70
 - recording client IP-to-MAC mapping (DHCP), 70
 - recording client IP-to-MAC mapping (DHCPv6), 169
 - relay agent support for Option 82 (DHCP), 56
 - snooping support for Option 82 (DHCP), 72
 - specifying server IP address for DHCP, 44
 - troubleshooting relay agent configuration (DHCP), 66
- command (troubleshooting IPv6 with ping), 148
- common address pool (DHCP), 35
- configuring
 - 6to4 tunnel, 185, 187
 - anycast address (IPv6), 130

- ARP, 1, 4, 7
- ARP snooping, 20
- automatic generation of link-local address (IPv6), 129
- basic IPv6, 117, 126, 143
- BOOTP client, 81, 82
- client (IPv4 DNS), 86
- client BIMS server information (DHCP), 42
- client DNS server (DHCP), 41
- client domain name suffix (DHCP), 41
- client gateway (DHCP), 43
- client NetBIOS node type (DHCP), 41
- client TFTP bootfile name (DHCP), 44
- client TFTP server (DHCP), 44
- client WINS server (DHCP), 41
- common address pool allocation mode (DHCP), 38
- DHCP client, 67
- DHCP packet rate limit, 78
- DHCP relay agent, 64
- DHCP server, 35, 50
- DHCP snooping, 70, 79
- DHCPv6 address pool, 155
- DHCPv6 client, 165
- DHCPv6 server, 153, 157
- DHCPv6 snooping, 168, 170
- DNS proxy, 88, 93
- DNS spoofing, 88
- dynamic address allocation mode (DHCP extended pool), 40
- dynamic address allocation mode (DHCP), 39
- dynamic domain name resolution (IPv4 DNS), 86, 90
- dynamic domain name resolution (IPv6 DNS), 97
- dynamic IP address assignment (DHCP), 51
- dynamic PMTU aging timer (IPv6), 138
- global unicast address (IPv6), 126
- gratuitous ARP, 11, 12
- GRE over IPv4 tunnel, 207, 209, 213
- GRE over IPv6 tunnel, 212
- GRE tunnel interface, 179
- ICMP to send error packet (IP performance optimization), 106, 107
- ICMPv6 packet sending (IPv6), 139
- interface max number snooping entries learned (DHCPv6), 170
- interface to dynamically obtain IP address (BOOTP), 82
- interface to generate EUI-64 IPv6 address (IPv6), 126
- IP address conflict detection (DHCP), 48
- IP addressing, 21, 23
- IP performance optimization, 103, 104
- IP unnumbered, 25
- IPv4 DNS, 83, 89
- IPv4 over IPv4 tunnel, 194, 195
- IPv4 over IPv6 tunnel, 198, 199
- IPv6 DNS, 95, 96
- IPv6 DNS client, 95
- IPv6 DNS dynamic domain name resolution, 95
- IPv6 DNS static domain name resolution, 95
- IPv6 manual tunnel, 180
- IPv6 over IPv6 tunnel, 203, 204
- IRDP, 110, 112, 113
- ISATAP tunnel, 189, 191
- isolate-user-VLAN local proxy ARP, 18
- link-local address (IPv6), 128
- link-local address manually (IPv6), 129
- max number dynamic interface entries (ARP table), 4
- max number dynamically learned neighbors (IPv6 NDP), 131

- max number ICMPv6 error packets sent interval (IPv6), 139
- max number NS DAD message send attempts (IPv6 NDP), 134
- multicast ARP, 9
- multicast ARP for NLB, 6
- periodic packet sending (gratuitous ARP), 11
- PMTU discovery (IPv6), 138
- port isolation local proxy ARP, 16
- proxy ARP, 13, 15
- quick update (ARP), 6
- RA message related parameter (IPv6), 133
- RA message-related parameter (IPv6), 131
- relay agent (DHCPv6), 161, 162, 163
- relay agent dynamic client entry periodic refresh (DHCP), 59
- relay agent Option 82 support (DHCP), 65
- relay agent security function (DHCP), 59
- relay agent support for Option 82 (DHCP), 62
- relay agent to release IP address (DHCP), 62
- self-defined option (DHCP), 45, 53
- sending RA message sending permission (IPv6), 132
- server address pool (DHCP), 37
- server security function (DHCP), 47
- snooping (IPv6 NDP), 135, 136
- snooping basic functions (DHCP), 74
- snooping entries backup (DHCP), 76
- snooping Option 82 support (DHCP), 80
- snooping support for Option 82 (DHCP), 75
- snooping trusted port (DHCPv6), 169
- stateless address autoconfiguration (IPv6), 127
- stateless DHCPv6, 151, 166
- static address allocation mode (DHCP), 38
- static domain name resolution (IPv4 DNS), 86, 89
- static domain name resolution (IPv6 DNS), 96
- static entry (ARP table), 4
- static IP address assignment (DHCP), 50
- static neighbor entry (IPv6 NDP), 130
- static PMTU for specified address (IPv6), 138
- super VLAN local proxy ARP, 17
- TCP attribute (IP performance optimization), 105
- TCP property (IPv6), 139
- TCP send/receive buffer size, 105
- TCP timers, 105
- tunneling, 172
- UDP Helper, 115, 116
- voice service client Option 184 parameter (DHCP), 43
- contacting HP, 219
- correlating server group with relay agent interface (DHCP), 58
- CPU (DHCP packet rate limit), 78
- creating
 - DHCP address pool, 38
 - DHCPv6 prefix pool, 155
 - static binding (DHCP), 59
- custom (DHCP), 31
- DAD message (IPv6 NDP), 134
- decapsulation
 - IPv4 over IPv4 tunneling, 175
 - IPv4/IPv6 over IPv6 tunneling, 176
- destination
 - address (RA), 111
 - unreachable message (IPv6), 141
- detecting
 - configuring IP address conflict detection (DHCP), 48
 - enabling relay agent offline detection (DHCP), 61
 - enabling relay agent unauthorized server detection (DHCP), 60
 - enabling unauthorized server detection (DHCP), 47

device

- ARP address resolution process, 2
- ARP configuration, 1, 4, 7, 13, 15
- ARP table, 3
- ARP table dynamic entry, 3, 4
- ARP table static entry, 3, 4
- common proxy ARP, 13
- DHCP overview, 27
- DHCP server configuration, 35
- enabling local proxy ARP, 15
- enabling proxy ARP, 14
- gratuitous ARP configuration, 11, 12
- IP addressing classes, 21
- IP addressing configuration, 21, 23
- local proxy ARP, 14
- masking (IP addressing), 22
- multicast ARP configuration, 9
- special IP addresses, 22
- subnetting (IP addressing), 22
- UDP Helper configuration, 115, 116

DHCP

- address pool, 35
- address pool selection, 36
- address pool structure, 35
- address pool type, 35
- application environment, 35
- applying extended address pool on interface, 47
- BOOTP client configuration, 81, 82
- client configuration, 67
- common options, 30
- configuring
 - relay agent to release IP address, 62
- configuring client BIMS server information, 42
- configuring client bootfile name, 44
- configuring client DNS server, 41

- configuring client domain name suffix, 41
- configuring client gateway, 43
- configuring client NetBIOS node type, 41
- configuring client TFTP server, 44
- configuring client WINS server, 41
- configuring common address pool allocation mode, 38
- configuring dynamic address allocation, 39
- configuring dynamic address allocation (extended pool), 40
- configuring dynamic IP address assignment, 51
- configuring IP address conflict detection, 48
- configuring packet rate limit, 78
- configuring relay agent dynamic client entry periodic refresh, 59
- configuring relay agent Option 82 support, 65
- configuring relay agent security function, 59
- configuring relay agent support for Option 82, 62
- configuring self-defined option, 45, 53
- configuring server address pool, 37
- configuring server security function, 47
- configuring snooping basic functions, 74
- configuring snooping entries backup, 76
- configuring snooping Option 82 support, 80
- configuring snooping support for Option 82, 75
- configuring static address allocation, 38
- configuring static IP address assignment, 50
- configuring voice service client Option 184 parameter, 43
- correlating server group with relay agent interface, 58
- creating address pool, 38
- creating static binding, 59
- custom options, 31
- DHCPv6. *See* DHCPv6
- displaying client, 67
- displaying relay agent, 64

- displaying server, 49
- displaying snooping, 78
- dynamic IP address allocation process, 28
- enabling, 46, 57
- enabling client on interface, 67
- enabling DHCP-REQUEST message attack protection, 77
- enabling Option 82 handling, 48
- enabling relay agent address check, 59
- enabling relay agent offline detection, 61
- enabling relay agent on interface, 58
- enabling relay agent starvation attack protection, 60
- enabling relay agent unauthorized server detection, 60
- enabling server on interface, 46
- enabling snooping starvation attack protection, 77
- enabling unauthorized server detection, 47
- ensuring client obtains authorized IP address from server, 70
- IP address allocation, 27
- IP address allocation mechanism, 27
- IP address allocation sequence, 36
- IP address lease extension, 28
- maintaining relay agent, 64
- maintaining server, 49
- maintaining snooping, 78
- message format, 29
- Option 184. *See* Option 184
- Option 82. *See* Option 82 (DHCP)
- options, 30
- overview, 27
- protocols and standards, 34
- recording client IP-to-MAC mapping, 70
- relay agent application environment, 55
- relay agent configuration, 55, 64
- relay agent option, 32
- relay agent support for Option 82, 56
- self-defined options, 31
- server configuration, 35, 50
- snooping configuration, 70, 79
- snooping function, 70
- snooping support for Option 82, 72
- specifying server IP address for client, 44
- specifying trap message send threshold, 49
- troubleshooting relay agent configuration, 66
- troubleshooting server configuration, 54
- trusted port application environment, 71
- vendor-specific option (Option 43). *See* Option 43 (DHCP)

DHCP-REQUEST, 77

DHCPv6

- address/prefix assignment, 149
- address/prefix lease renewal, 150
- client configuration, 165
- configuring interface max number snooping entries learned, 170
- configuring relay agent, 161, 162, 163
- configuring server, 153, 157
- configuring snooping trusted port, 169
- creating address pool, 155
- creating prefix pool, 155
- displaying client, 165
- displaying relay agent, 163
- displaying server, 157
- displaying snooping, 170
- DUID, 153
- enabling server, 155
- enabling snooping, 169
- ensuring client obtains address from authorized server, 168
- four message assignment, 150

- IA, 154
- maintaining client, 165
- maintaining relay agent, 163
- maintaining server, 157
- maintaining snooping, 170
- multicast address, 153
- overview, 149
- PD, 154
- prefix selection process, 154
- recording client IP-to-MAC mapping, 169
- relay agent application environment, 161
- relay agent operation, 161
- server application environment, 153
- snooping configuration, 168, 170
- stateless configuration, 151, 166
- stateless operation, 151
- stateless protocols and standards, 152
- two message rapid assignment, 149
- discovery
 - configuring PMTU discovery (IPv6), 138
 - neighbor. See NDP
- displaying
 - ARP, 7
 - ARP snooping, 20
 - BOOTP client configuration, 82
 - DHCP client, 67
 - DHCP server, 49
 - DHCPv6 client, 165
 - DHCPv6 server, 157
 - DHCPv6 snooping, 170
 - IP addressing, 26
 - IP performance optimization, 109
 - IPv4 DNS, 89
 - IPv6 basic configuration, 141
 - IPv6 DNS, 96
 - proxy ARP, 15
 - relay agent (DHCP), 64
 - relay agent (DHCPv6), 163
 - snooping (DHCP), 78
 - tunneling configuration, 217
 - UDP Helper, 116
- DNS
 - configuring client DNS server (DHCP), 41
 - configuring IPv4 client, 86
 - configuring IPv4 dynamic domain name resolution, 86, 90
 - configuring IPv4 static domain name resolution, 86, 89
 - configuring IPv6 DNS client, 95
 - configuring IPv6 dynamic domain name resolution, 97
 - configuring IPv6 static domain name resolution, 96
 - configuring proxy, 88
 - configuring spoofing, 88
 - displaying IPv4, 89
 - displaying IPv6, 96
 - dynamic domain name resolution (IPv4), 83
 - dynamic domain name resolution (IPv6), 95
 - dynamic domain name resolution process, 83
 - IPv4 configuration, 83, 89
 - IPv4 proxy configuration, 93
 - IPv6 configuration, 95, 96
 - maintaining IPv4, 89
 - maintaining IPv6, 96
 - proxy, 84
 - proxy operation, 85
 - spoofing, 85
 - static domain name resolution (IPv4), 83
 - static domain name resolution (IPv6), 95
 - suffixes, 84
- documentation

- conventions used, 220
- website, 219
- domain
 - configuring client DNS server (DHCP), 41
 - configuring client domain name suffix (DHCP), 41
 - configuring client NetBIOS node type (DHCP), 41
 - configuring dynamic domain name resolution (IPv4 DNS), 86, 90
 - configuring dynamic domain name resolution (IPv6 DNS), 97
 - configuring static domain name resolution (IPv4 DNS), 86, 89
 - configuring static domain name resolution (IPv6 DNS), 96
 - dynamic name resolution (IPv4 DNS), 83
 - dynamic name resolution (IPv6 DNS), 95
 - static name resolution (IPv4 DNS), 83
 - static name resolution (IPv6 DNS), 95
- DUID (DHCPv6 identifier), 153
- duplicate address detection (IPv6 NDP), 121
- dynamic
 - ARP table entry, 3, 4
 - DHCP IP address allocation process, 28
 - domain name resolution (DNS), 83
 - domain name resolution (IPv6 DNS), 95
 - obtaining address dynamically (BOOTP), 81
- Dynamic Host Configuration Protocol. *See* DHCP
- echo requests (IPv6 multicast), 140
- enabling
 - client on interface (DHCP), 67
 - DHCP, 46
 - DHCP-REQUEST message attack protection (DHCP), 77
 - DHCPv6 server, 155
 - dynamic ARP entry check, 5
 - forwarding of broadcasts to network (IP performance optimization), 103
 - ICMP extension support, 107, 108
 - ICMPv6 destination unreachable message send (IPv6), 141
 - ICMPv6 time exceeded packet sending (IPv6), 140
 - IPv6, 126
 - local ND proxy, 137
 - ND proxy, 136, 137
 - Option 82 handling (DHCP), 48
 - packet learning (gratuitous ARP), 11
 - reception of broadcasts by network (IP performance optimization), 103
 - relay agent address check (DHCP), 59
 - relay agent offline detection (DHCP), 61
 - relay agent starvation attack protection (DHCP), 60
 - relay agent unauthorized server detection (DHCP), 60
 - replying to multicast echo requests (IPv6), 140
 - server on interface (DHCP), 46
 - snooping (DHCPv6), 169
 - snooping starvation attack protection (DHCP), 77
 - unauthorized server detection (DHCP), 47
- encapsulation
 - GRE de-encapsulation process, 178
 - GRE packet format, 177
 - GRE process, 177
 - IPv4 over IPv4 tunneling, 175
 - IPv4 over IPv4 tunneling decapsulation, 175
 - IPv4/IPv6 over IPv6 tunneling, 176
 - IPv4/IPv6 over IPv6 tunneling decapsulation, 176
 - tunneling configuration, 172
- entry check (dynamic ARP), 5
- error packets (ICMPv6), 139
- Ethernet
 - BOOTP client configuration, 81, 82
 - DHCP overview, 27
 - DHCP server configuration, 35

- dynamically obtain address (BOOTP), 81
- enabling local proxy ARP in interface view, 15
- enabling proxy ARP in interface view, 14
- IP performance optimization configuration, 103, 104
- IPv6 basic configuration, 117, 126, 143
- IPv6 NDP configuration, 130
- TCP attribute configuration (IP performance optimization), 105
- UDP Helper configuration, 115, 116

EUI-64

- address-based interface identifier (IPv6), 120
- configuring interface to generate address (IPv6), 126
- specifying
 - interface IPv6 address manually (IPv6), 127

extended address pool (DHCP), 35, 40

extension

- enabling support for ICMP extensions, 108
- ICMP extensions for MPLS, 107
- ICMP support, 107

feature

- address autoconfiguration (IPv6), 117
- address space (IPv6), 117
- built-in security (IPv6), 118
- flexible extension header (IPv6), 118
- header format simplification (IPv6), 117
- hierarchical address structure (IPv6), 117
- IPv6, 117
- neighbor discovery mechanism (IPv6), 118
- QoS support (IPv6), 118

field (DHCP option), 30

file (DHCP bootfile), 44

format

- address (IPv6), 118
- GRE packet encapsulation, 177
- header format simplification (IPv6), 117
- message (ARP), 1
- message (DHCP), 29

four message assignment (DHCPv6), 150

function

- ARP, 1
- snooping (DHCP), 70

gateway (DHCP configuration), 43

Generic Routing Encapsulation. See GRE

global address (IPv6 unicast), 119

gratuitous ARP

- configuration, 11, 12
- configuring periodic packet sending, 11
- enabling packet learning, 11

GRE

- configuring IPv6 manual tunnel, 180
- configuring over IPv4 tunnel, 207, 209, 213
- configuring over IPv6 tunnel, 212
- configuring tunnel interface, 179
- de-encapsulation process, 178
- encapsulation process, 177
- packet encapsulation format, 177
- tunneling discussion, 177
- tunneling protocols and standards, 178

handling ICMP messages, 108

header

- flexible extension (IPv6), 118
- format (IPv6), 117

HP

- customer support and resources, 219
- document conventions, 220
- documents and manuals, 219
- icons used, 220
- subscription service, 219
- support contact information, 219

- symbols used, 220
- websites, 219
- ICMP
 - enabling extension support, 107
 - enabling support for extensions, 108
 - error packet (IP performance optimization), 106, 107
 - extensions for MPLS, 107
 - handling messages, 108
 - IRDP configuration, 110, 112, 113
- ICMPv6
 - address autoconfiguration (IPv6 NDP), 123
 - address resolution (IPv6), 122
 - configuring max number error packets sent interval (IPv6), 139
 - configuring packet sending, 139
 - duplicate address detection (IPv6 NDP), 122
 - enabling destination unreachable message send (IPv6), 141
 - enabling ICMPv6 time exceeded packet sending (IPv6), 140
 - enabling replying to multicast echo requests (IPv6), 140
 - messages used to implement NDP, 121
 - NDP configuration (IPv6), 130
 - PMTU discovery (IPv6), 124
 - redirection (IPv6 NDP), 123
 - router/prefix discovery (IPv6 NDP), 123
 - transition technologies (IPv6), 124
- icons, 220
- identifier
 - DHCPv6 IAID, 154
- identifier
 - DHCPv6 DUID, 153
 - EUI-64, 120
- interface
 - configuring automatic generation of link-local address (IPv6), 129
 - configuring to dynamically obtain IP address (BOOTP), 82
- IP
 - configuring ICMP to send error packet (performance optimization), 106, 107
 - GRE tunneling, 177
 - performance optimization. See IP performance optimization
 - tunneling configuration, 172
- IP addressing
 - address allocation (DHCP), 27
 - address allocation mechanism (DHCP), 27
 - address allocation sequence (DHCP), 36
 - address lifetime, 111
 - address preference, 111
 - address/prefix assignment (DHCPv6), 149
 - address/prefix lease renewal (DHCPv6), 150
 - applying extended address pool on interface (DHCP), 47
 - assigning address to interface, 23
 - class, 21
 - configuration, 21, 23
 - configuring address pool allocation mode (DHCP), 38
 - configuring client BIMS server information (DHCP), 42
 - configuring client bootfile name (DHCP), 44
 - configuring client DNS server (DHCP), 41
 - configuring client domain name suffix (DHCP), 41
 - configuring client gateway (DHCP), 43
 - configuring client NetBIOS node type (DHCP), 41
 - configuring client TFTP server (DHCP), 44
 - configuring client WINS server (DHCP), 41
 - configuring DHCPv6 server, 153, 157
 - configuring dynamic address allocation (DHCP extended pool), 40

- configuring dynamic address allocation (DHCP), 39
- configuring dynamic IP address assignment (DHCP), 51
- configuring interface max number snooping entries learned (DHCPv6), 170
- configuring interface to dynamically obtain address (BOOTP), 82
- configuring IP address conflict detection (DHCP), 48
- configuring IP unnumbered, 25
- configuring relay agent (DHCPv6), 161, 162, 163
- configuring relay agent dynamic client entry periodic refresh (DHCP), 59
- configuring relay agent security function (DHCP), 59
- configuring relay agent support for Option 82 (DHCP), 62
- configuring relay agent to release IP address (DHCP), 62
- configuring self-defined option (DHCP), 45, 53
- configuring server address pool (DHCP), 37
- configuring server security function (DHCP), 47
- configuring snooping basic functions (DHCP), 74
- configuring snooping entries backup 82 (DHCP), 76
- configuring snooping support for Option 82 (DHCP), 75
- configuring snooping trusted port (DHCPv6), 169
- configuring stateless DHCPv6, 166
- configuring static address allocation (DHCP), 38
- configuring static IP address assignment (DHCP), 50
- configuring voice service client Option 184 parameter (DHCP), 43
- correlating server group with relay agent interface (DHCP), 58
- creating address pool (DHCP), 38
- creating static binding (DHCP), 59
- DHCP client configuration, 67
- DHCP overview, 27
- DHCP relay agent configuration, 55, 64
- DHCP server configuration, 35, 50
- DHCP snooping configuration), 70, 79
- DHCPv6 client configuration, 165
- DHCPv6 overview, 149
- DHCPv6 snooping configuration, 168, 170
- displaying, 26
- dynamic allocation process (DHCP), 28
- dynamically obtain address (BOOTP), 81
- enabling client on interface (DHCP), 67
- enabling DHCP, 46, 57
- enabling DHCP-REQUEST message attack protection (DHCP), 77
- enabling Option 82 handling (DHCP), 48
- enabling relay agent address check (DHCP), 59
- enabling relay agent offline detection (DHCP), 61
- enabling relay agent on interface (DHCP), 58
- enabling relay agent starvation attack protection (DHCP), 60
- enabling relay agent unauthorized server detection (DHCP), 60
- enabling server on interface (DHCP), 46
- enabling snooping (DHCPv6), 169
- enabling snooping starvation attack protection (DHCP), 77
- enabling unauthorized server detection (DHCP), 47
- ensuring client obtains address from authorized server (DHCPv6), 168
- ensuring client obtains authorized IP address from server (DHCP), 70
- four message assignment (DHCPv6), 150
- IPv4 DNS troubleshooting configuration, 94
- IPv6 basic configuration, 117, 126, 143
- IRDP configuration, 110, 112, 113
- lease extension (DHCP), 28
- masking, 22

- proxy-advertised IP addresses, 111
 - recording client IP-to-MAC mapping (DHCP), 70
 - recording client IP-to-MAC mapping (DHCPv6), 169
 - relay agent application environment (DHCP), 55
 - relay agent application environment (DHCPv6), 161
 - relay agent operation (DHCPv6), 161
 - relay agent support for Option 82 (DHCP), 56
 - server application environment (DHCPv6), 153
 - snooping function (DHCP), 70
 - snooping support for Option 82 (DHCP), 72
 - special addresses, 22
 - specifying server address for DHCP client, 44
 - specifying trap message send threshold (DHCP), 49
 - stateless configuration (DHCPv6), 151
 - stateless operation (DHCPv6), 151
 - subnetting, 22
 - troubleshooting DHCP server configuration, 54
 - troubleshooting IPv6 basic configuration, 148
 - trusted port application environment (DHCP), 71
 - two message rapid assignment (DHCPv6), 149
- IP performance optimization
- configuration, 103, 104
 - configuring ICMP to send error packet, 106, 107
 - displaying, 109
 - enabling forwarding of broadcasts to network, 103
 - enabling reception of broadcasts by network, 103
 - maintaining, 109
 - TCP attribute configuration, 105
- IPng. See IPv6
- IPv4
- configuring 6to4 tunnel, 185, 187
 - configuring client (DNS), 86
 - configuring dynamic domain name resolution (DNS), 86, 90
 - configuring GRE over IPv4 tunnel, 207, 209, 213
 - configuring IPv4 over IPv4 tunnel, 194, 195
 - configuring IPv4 over IPv6 tunnel, 198, 199
 - configuring static domain name resolution (DNS), 86, 89
 - DNS configuration, 83, 89
 - DNS proxy configuration, 93
 - dynamic domain name resolution (DNS), 83
 - IPv4/IPv6 over IPv6 tunneling, 176
 - IPv4/IPv6 over IPv6 tunneling decapsulation, 176
 - IPv4/IPv6 over IPv6 tunneling encapsulation, 176
 - IPv4/IPv6 tunnels, 172
 - IPv6 over IPv4 automatic tunnel, 173
 - IPv6 over IPv4 manually configured tunnel, 173
 - IPv6 over IPv4 tunneling, 173
 - over IPv4 tunneling, 175
 - over IPv4 tunneling decapsulation, 175
 - over IPv4 tunneling encapsulation, 175
 - static domain name resolution (DNS), 83
 - tunneling types, 174
- IPv4 DNS
- configuration, 83, 89
 - configuring client, 86
 - configuring dynamic domain name resolution, 86, 90
 - configuring static domain name resolution, 86, 89
 - dynamic domain name resolution, 83
 - proxy configuration, 93
 - static domain name resolution, 83
 - troubleshooting configuration, 94
- IPv6
- address, 118
 - address autoconfiguration (NDP), 123
 - address format, 118

- address resolution, 122
- address space, 117
- address structure, 117
- address type, 119
- address/prefix assignment (DHCPv6), 149
- address/prefix lease renewal (DHCPv6), 150
- allowing RA message sending, 132
- anycast address type, 119
- automatic address configuration, 117
- basic configuration, 117, 126, 143
- configuring 6to4 tunnel, 185, 187
- configuring anycast address, 130
- configuring automatic generation of link-local address, 129
- configuring DHCPv6 server, 153, 157
- configuring dynamic domain name resolution (DNS), 97
- configuring dynamic PMTU aging timer, 138
- configuring global unicast address, 126
- configuring GRE over IPv6 tunnel, 212
- configuring ICMPv6 packet sending, 139
- configuring interface max number snooping entries learned (DHCPv6), 170
- configuring interface to generate EUI-64 IPv6 address, 126
- configuring IPv4 over IPv6 tunnel, 198, 199
- configuring IPv6 over IPv6 tunnel, 203, 204
- configuring link-local address, 128
- configuring link-local address manually, 129
- configuring manual tunnel, 180
- configuring max number dynamically learned neighbors, 131
- configuring max number ICMPv6 error packets sent interval, 139
- configuring max number NS DAD message send attempts, 134
- configuring NDP snooping, 135, 136
- configuring PMTU discovery, 138
- configuring RA message related parameter, 133
- configuring RA message-related parameter, 131
- configuring relay agent (DHCPv6), 161, 162, 163
- configuring snooping trusted port (DHCPv6), 169
- configuring stateless address autoconfiguration, 127
- configuring stateless DHCPv6, 166
- configuring static domain name resolution (DNS), 96
- configuring static neighbor entry, 130
- configuring static PMTU for specified address, 138
- configuring TCP property, 139
- DHCPv6 client configuration, 165
- DHCPv6 overview, 149
- DHCPv6 snooping configuration, 168, 170
- displaying basic configuration, 141
- DNS configuration, 95, 96
- duplicate address detection (NDP), 122
- dynamic domain name resolution (DNS), 95
- enabling, 126
- enabling ICMPv6 destination unreachable message send, 141
- enabling ICMPv6 time exceeded packet sending, 140
- enabling local ND proxy, 137
- enabling ND proxy, 136, 137
- enabling replying to multicast echo requests, 140
- enabling snooping (DHCPv6), 169
- ensuring client obtains address from authorized server (DHCPv6), 168
- EUI-64 address-based interface identifier, 120
- features, 117
- flexible extension header, 118
- four message assignment (DHCPv6), 150
- header format simplification, 117
- IPv4/IPv6 tunnels, 172

- maintaining basic configuration, 141
- multicast address, 120
- multicast address type, 119
- NDP, 121
- NDP configuration, 130
- neighbor discovery mechanism, 118
- neighbor reachability detection (NDP), 122
 - over IPv4 automatic tunnel, 173
 - over IPv4 manually configured tunnel, 173
 - over IPv4 tunneling, 173
- PMTU discovery, 124
- QoS support, 118
- recording client IP-to-MAC mapping (DHCPv6), 169
- redirection(NDP), 123
- relay agent application environment (DHCPv6), 161
- relay agent operation (DHCPv6), 161
- router/prefix discovery (NDP), 123
- security, 118
- server application environment (DHCPv6), 153
- specifying interface EUI-64 IPv6 address manually, 127
- stateless configuration (DHCPv6), 151
- stateless operation (DHCPv6), 151
- static domain name resolution (DNS), 95
- transition technologies, 124
- troubleshooting basic configuration, 148
- tunneling types, 174
- two message rapid assignment (DHCPv6), 149
- unicast address, 119
- unicast address type, 119

IPv6 DNS

- client configuration, 95
- configuration, 95, 96
- configuring dynamic domain name resolution, 97
 - configuring static domain name resolution, 96
 - dynamic domain name resolution, 95
 - static domain name resolution, 95

IPv6 NDP protocols and standards, 125

IRDP

- advertising interval, 111
- configuration, 110, 112, 113
- IP address lifetime, 111
- IP address preference, 111
- protocols and standards, 111
- proxy-advertised IP addresses, 111
- RA destination address, 111
- terminology, 111
- working mechanism, 110

ISATAP tunnel configuration, 189, 191

isolate-user-VLAN (ARP configuration), 18

Layer 2 ARP snooping configuration), 20

Layer 3

- enabling local proxy ARP in Ethernet interface view, 15
- enabling proxy ARP in Ethernet interface view, 14

learning (gratuitous ARP), 11

lease

- extension (DHCP), 28
- record (DHCPv6 PD), 154
- renewal (DHCPv6), 150

lifetime (IP address), 111

link-local

- address (IPv6 unicast), 119
- address (IPv6), 128
- address autogeneration for interface (IPv6), 129
- address manual configuration (IPv6), 129
- configuring anycast address (IPv6), 130

local proxy ARP, 14

loopback address (IPv6 unicast), 119

- MAC address
 - ARP table, 3
 - ARP table dynamic entry, 3, 4
 - ARP table static entry, 3, 4
 - BOOTP client configuration, 81, 82
 - recording client IP-to-MAC mapping (DHCP), 70
 - recording client IP-to-MAC mapping (DHCPv6), 169
- maintaining
 - ARP, 7
 - ARP snooping, 20
 - DHCP server, 49
 - DHCPv6 client, 165
 - DHCPv6 server, 157
 - DHCPv6 snooping, 170
 - IP performance optimization, 109
 - IPv4 DNS, 89
 - IPv6 basic configuration, 141
 - IPv6 DNS, 96
 - proxy ARP, 15
 - relay agent (DHCP), 64
 - relay agent (DHCPv6), 163
 - snooping (DHCP), 78
 - tunneling configuration, 217
 - UDP Helper, 116
- manuals, 219
- mapping (IP-to-MAC), 70
- masking (IP addressing), 22
- maximum number dynamically learned neighbors (IPv6 NDP), 131
- message
 - allowing RA message sending (IPv6), 132
 - configuring max number NS DAD message send attempts (IPv6), 134
 - configuring RA message related parameter (IPv6), 133
 - configuring RA message-related parameter (IPv6), 131
 - enabling DHCP-REQUEST message attack protection (DHCP), 77
 - enabling ICMP extension support, 107
 - enabling ICMPv6 destination unreachable message send (IPv6), 141
 - format (ARP), 1
 - format (DHCP), 29
 - handling ICMP messages, 108
 - specifying trap message send threshold (DHCP), 49
- mode
 - configuring common address pool allocation mode (DHCP), 38
 - configuring dynamic address allocation mode (DHCP extended pool), 40
 - configuring dynamic address allocation mode (DHCP), 39
 - configuring static address allocation mode (DHCP), 38
 - relay agent (DHCPv6), 161
 - stateless DHCPv6, 151
- MPLS (ICMP extension), 107
- multicast
 - address (IPv6), 120
 - ARP configuration, 9
 - DHCPv6 multicast address, 153
 - enabling replying to multicast echo requests (IPv6), 140
- NDP
 - address autoconfiguration (IPv6), 123
 - address resolution (IPv6), 122
 - configuring max number dynamically learned neighbors (IPv6), 131
 - configuring snooping (IPv6), 135, 136
 - configuring static neighbor entry (IPv6), 130
 - duplicate address detection (IPv6), 122

- enabling local ND proxy, 137
- enabling ND proxy, 136, 137
- IPv6, 121
- IPv6 configuration, 130
- mechanism (IPv6), 118
- neighbor reachability detection (IPv6), 122
- redirection (IPv6), 123
- router/prefix discovery (IPv6), 123
- setting age timer for ND entries, 134
- neighbor discovery. *See* NDP
- Neighbor Discovery Protocol. *See* NDP
- neighbor reachability detection (IPv6 NDP), 121
- NetBIOS (DHCP), 41
- network
 - enabling forwarding of broadcasts to network (IP performance optimization), 103
 - enabling reception of broadcasts by network (IP performance optimization), 103
- network management
 - 6to4 tunnel configuration, 187
 - ARP configuration, 1, 4, 7
 - ARP snooping configuration), 20
 - BOOTP client configuration, 81, 82
 - common proxy ARP, 13
 - configuring relay agent (DHCPv6), 163
 - DHCP client configuration, 67
 - DHCP overview, 27
 - DHCP relay agent configuration, 55, 64
 - DHCP server configuration, 35
 - DHCP snooping configuration), 70, 79
 - DHCP snooping Option 82 support configuration, 80
 - DHCPv6 client configuration, 165
 - DHCPv6 overview, 149
 - DNS
 - spoofing, 85
 - DNS proxy configuration, 93
 - dynamic domain name resolution configuration (IPv4 DNS), 90
 - dynamic IP address assignment configuration (DHCP), 51
 - enabling local proxy ARP, 15
 - enabling proxy ARP, 14
 - gratuitous ARP configuration, 11, 12
 - IP addressing configuration, 21, 23
 - IP performance optimization configuration, 103, 104
 - IPv4 DNS configuration, 83, 89
 - IPv6 basic configuration, 117, 126, 143
 - IPv6 DNS client configuration, 95
 - IPv6 DNS configuration, 95, 96
 - IPv6 DNS dynamic domain name resolution configuration, 97
 - IPv6 DNS static domain name resolution configuration, 96
 - IPv6 NDP configuration, 130
 - IRDP configuration, 110, 112, 113
 - ISATAP tunnel configuration, 189, 191
 - isolate-user-VLAN local proxy ARP configuration, 18
 - local proxy ARP, 14
 - multicast ARP configuration, 9
 - port isolation proxy ARP configuration, 16
 - proxy ARP configuration, 13, 15
 - relay agent Option 82 support configuration (DHCP), 65
 - self-defined option configuration (DHCP), 53
 - stateless DHCPv6 configuration, 151
 - static domain name resolution configuration (IPv4 DNS), 89
 - static IP address assignment configuration (DHCP), 50
 - super VLAN local proxy ARP configuration, 17

- TCP attribute configuration (IP performance optimization), 105
- tunneling configuration, 172
- UDP Helper configuration, 115, 116
- networking
 - GRE tunneling, 177
 - masking (IP addressing), 22
 - subnetting (IP addressing), 22
- NLB (multicast ARP), 6
- NS DAD message (IPv6), 134
- offline detection (DHCP), 61
- optimizing IP performance, 103, 104
- option
 - common (DHCP), 30
 - configuring relay agent (DHCP), 55, 64
 - configuring relay agent dynamic client entry periodic refresh (DHCP), 59
 - configuring relay agent security function (DHCP), 59
 - configuring relay agent support for Option 82 (DHCP), 62
 - configuring self-defined option (DHCP), 53
 - configuring snooping support for Option 82 (DHCP), 75
 - correlating server group with relay agent interface (DHCP), 58
 - creating static binding (DHCP), 59
 - custom (DHCP), 31
 - enabling Option 82 handling (DHCP), 48
 - enabling relay agent address check (DHCP), 59
 - enabling relay agent on interface (DHCP), 58
 - field (DHCP), 30
 - relay agent application environment (DHCP), 55
 - relay agent support for Option 82 (DHCP), 56
 - self-defined (DHCP), 31
 - snooping support for Option 82 (DHCP), 72
- Option 121 (DHCP), 30
- Option 150 (DHCP), 30
- Option 184
 - configuring voice service client Option 184 parameter (DHCP), 43
 - DHCP reserved, 33
- Option 3 (DHCP), 30
- Option 33 (DHCP), 30
- Option 43 (DHCP), 31
- Option 51 (DHCP), 30
- Option 53 (DHCP), 30
- Option 55 (DHCP), 30
- Option 6 (DHCP), 30
- Option 60 (DHCP), 30
- Option 66 (DHCP), 30
- Option 82
 - configuring relay agent security function (DHCP), 59
 - configuring relay agent support (DHCP), 62
 - configuring snooping support (DHCP), 75, 80
 - correlating server group with relay agent interface (DHCP), 58
 - DHCP client location, 32
 - enabling (DHCP), 48
 - enabling DHCP, 57
 - enabling relay agent on interface (DHCP), 58
 - relay agent application environment (DHCP), 55
 - relay agent configuration (DHCP), 55, 64
 - relay agent support (DHCP), 56
 - snooping support (DHCP), 72
- packet
 - ARP configuration, 13, 15
 - configuring DHCP packet rate limit, 78
 - configuring ICMP to send error packet (IP performance optimization), 106, 107
 - configuring ICMPv6 sending (IPv6), 139
 - configuring max number ICMPv6 error packets sent interval (IPv6), 139

- configuring periodic sending (gratuitous ARP), 11
- enabling ICMPv6 time exceeded sending (IPv6), 140
- enabling learning (gratuitous ARP), 11
- gratuitous ARP configuration, 11, 12
- GRE de-encapsulation process, 178
- GRE encapsulation format, 177
- GRE encapsulation process, 177
- GRE tunneling, 177
- IPv4 over IPv4 tunneling, 175
- IPv4 over IPv4 tunneling decapsulation, 175
- IPv4 over IPv4 tunneling encapsulation, 175
- IPv4/IPv6 over IPv6 tunneling, 176
- IPv4/IPv6 over IPv6 tunneling decapsulation, 176
- IPv4/IPv6 over IPv6 tunneling encapsulation, 176
- IPv6 over IPv4 automatic tunnel, 173
- IPv6 over IPv4 manually configured tunnel, 173
- IPv6 over IPv4 tunneling, 173
- tunneling configuration, 172
- UDP Helper configuration, 115, 116

parameter

- configuring RA message related parameter (IPv6), 133
- configuring RA message-related parameter (IPv6), 131

path MTU. See PMTU

PD (DHCPv6), 154

periodic packet sending (gratuitous ARP), 11

ping (IPv6 troubleshooting), 148

PMTU

- configuring discovery (IPv6), 138
- configuring dynamic PMTU aging timer (IPv6), 138
- configuring static PMTU for specified address (IPv6), 138
- discovery (IPv6), 124

pool

- address (DHCP), 35
- address pool selection (DHCP), 36
- address pool structure (DHCP), 35
- address pool type (DHCP), 35
- applying DHCPv6 address pool to an interface, 156
- applying extended address pool on interface (DHCP), 47
- configuring common address pool allocation mode (DHCP), 38
- configuring dynamic address allocation for extended address pool (DHCP), 40
- configuring server address pool (DHCP), 37
- creating address pool (DHCP), 38
- DHCPv6 address, 155
- DHCPv6 prefix, 155

port

- BOOTP client configuration, 81, 82
- configuring snooping trusted port (DHCPv6), 169
- DHCP overview, 27
- DHCP server configuration, 35
- IP performance optimization configuration, 103, 104
- IPv6 basic configuration, 117, 126, 143
- IPv6 NDP configuration, 130
- isolation (local proxy ARP configuration, 16
- TCP attribute configuration (IP performance optimization), 105
- trusted port application environment (DHCP), 71
- UDP Helper configuration, 115, 116

preference (IP address), 111

prefix

- address/prefix assignment (DHCPv6), 149
- address/prefix lease renewal (DHCPv6), 150
- creating pool (DHCPv6), 155
- four message assignment (DHCPv6), 150
- selection process (DHCPv6), 154

- two message rapid assignment (DHCPv6), 149
- procedure
 - applying DHCPv6 address pool to an interface, 156
 - applying extended address pool on interface (DHCP), 47
 - assigning IP address to interface, 23
 - configuring 6to4 tunnel, 185, 187
 - configuring anycast address (IPv6), 130
 - configuring ARP snooping, 20
 - configuring automatic generation of link-local address (IPv6), 129
 - configuring basic IPv6, 143
 - configuring BOOTP client, 82
 - configuring client (IPv4 DNS), 86
 - configuring client BIMS server information (DHCP), 42
 - configuring client bootfile name (DHCP), 44
 - configuring client DNS server (DHCP), 41
 - configuring client domain name suffix (DHCP), 41
 - configuring client gateway (DHCP), 43
 - configuring client NetBIOS node type (DHCP), 41
 - configuring client TFTP server (DHCP), 44
 - configuring client WINS server (DHCP), 41
 - configuring common address pool allocation mode (DHCP), 38
 - configuring DHCP client, 67
 - configuring DHCP packet rate limit, 78
 - configuring DHCP relay agent, 64
 - configuring DHCP server, 35, 50
 - configuring DHCP snooping, 70, 79
 - configuring DHCPv6 address pool, 155
 - configuring DHCPv6 client, 165
 - configuring DHCPv6 server, 153, 157
 - configuring DHCPv6 snooping, 168, 170
 - configuring DNS proxy, 88, 93
 - configuring DNS spoofing, 88
 - configuring dynamic address allocation mode (DHCP extended pool), 40
 - configuring dynamic address allocation mode (DHCP), 39
 - configuring dynamic domain name resolution (IPv4 DNS), 86, 90
 - configuring dynamic domain name resolution (IPv6 DNS), 97
 - configuring dynamic IP address assignment (DHCP), 51
 - configuring dynamic PMTU aging timer (IPv6), 138
 - configuring global unicast address (IPv6), 126
 - configuring gratuitous ARP, 11, 12
 - configuring GRE over IPv4 tunnel, 207, 209, 213
 - configuring GRE over IPv6 tunnel, 212
 - configuring GRE tunnel interface, 179
 - configuring ICMP to send error packet (IP performance optimization), 106, 107
 - configuring ICMPv6 packet sending (IPv6), 139
 - configuring interface max number snooping entries learned (DHCPv6), 170
 - configuring interface to dynamically obtain IP address (BOOTP), 82
 - configuring interface to generate EUI-64 IPv6 address (IPv6), 126
 - configuring IP address conflict detection (DHCP), 48
 - configuring IP addressing, 23
 - configuring IP unnumbered, 25
 - configuring IPv4 over IPv4 tunnel, 194, 195
 - configuring IPv4 over IPv6 tunnel, 198, 199
 - configuring IPv6 DNS client, 95
 - configuring IPv6 DNS dynamic domain name resolution, 95
 - configuring IPv6 DNS static domain name resolution, 95
 - configuring IPv6 manual tunnel, 180
 - configuring IPv6 over IPv6 tunnel, 203, 204
 - configuring IRDP, 110, 112, 113

- configuring ISATAP tunnel, 189, 191
- configuring link-local address (IPv6), 128
- configuring link-local address manually (IPv6), 129
- configuring max number dynamic interface entries (ARP table), 4
- configuring max number dynamically learned neighbors (IPv6 NDP), 131
- configuring max number ICMPv6 error packets sent interval (IPv6), 139
- configuring max number NS DAD message send attempts (IPv6 NDP), 134
- configuring multicast ARP for NLB, 6
- configuring PMTU discovery (IPv6), 138
- configuring proxy ARP, 13, 15
- configuring quick update (ARP), 6
- configuring RA message related parameter (IPv6), 133
- configuring RA message sending permission (IPv6), 132
- configuring RA message-related parameter (IPv6), 131
- configuring relay agent (DHCPv6), 161, 162, 163
- configuring relay agent dynamic client entry periodic refresh (DHCP), 59
- configuring relay agent Option 82 support (DHCP), 65
- configuring relay agent security function (DHCP), 59
- configuring relay agent support for Option 82 (DHCP), 62
- configuring relay agent to release IP address (DHCP), 62
- configuring self-defined option (DHCP), 45, 53
- configuring server address pool (DHCP), 37
- configuring server security function (DHCP), 47
- configuring snooping (IPv6 NDP), 135, 136
- configuring snooping basic functions (DHCP), 74
- configuring snooping entries backup (DHCP), 76
- configuring snooping Option 82 support (DHCP), 80
- configuring snooping support for Option 82 (DHCP), 75
- configuring snooping trusted port (DHCPv6), 169
- configuring stateless address autoconfiguration (IPv6), 127
- configuring stateless DHCPv6, 166
- configuring static address allocation mode (DHCP), 38
- configuring static domain name resolution (IPv4 DNS), 86, 89
- configuring static domain name resolution (IPv6 DNS), 96
- configuring static entry (ARP table), 4
- configuring static IP address assignment (DHCP), 50
- configuring static neighbor entry (IPv6 NDP), 130
- configuring static PMTU for specified address (IPv6), 138
- configuring TCP property (IPv6), 139
- configuring TCP send/receive buffer size, 105
- configuring TCP timers, 105
- configuring tunneling, 172
- configuring UDP Helper, 115, 116
- configuring voice service client Option 184 parameter (DHCP), 43
- correlating server group with relay agent interface (DHCP), 58
- creating address pool (DHCP), 38
- creating DHCPv6 prefix pool, 155
- creating static binding (DHCP), 59
- displaying ARP, 7
- displaying ARP snooping, 20
- displaying BOOTP client configuration, 82
- displaying DHCP client, 67
- displaying DHCP server, 49
- displaying DHCP snooping, 78

displaying DHCPv6 client, 165
 displaying DHCPv6 relay agent, 163
 displaying DHCPv6 server, 157
 displaying DHCPv6 snooping, 170
 displaying IP addressing, 26
 displaying IP performance optimization, 109
 displaying IPv4 DNS, 89
 displaying IPv6 basic configuration, 141
 displaying IPv6 DNS, 96
 displaying proxy ARP, 15
 displaying relay agent (DHCP), 64
 displaying tunneling configuration, 217
 displaying UDP Helper, 116
 enabling ARP snooping on VLAN, 20
 enabling client on interface (DHCP), 67
 enabling DHCP, 46
 enabling DHCP server to handle Option 82, 48
 enabling DHCP-REQUEST message attack protection (DHCP), 77
 enabling DHCPv6 server, 155
 enabling dynamic ARP entry check, 5
 enabling forwarding of broadcasts to network (IP performance optimization), 103
 enabling forwarding of directed broadcast to network (IP performance optimization), 104
 enabling ICMP extension support, 107
 enabling ICMPv6 destination unreachable message send (IPv6), 141
 enabling ICMPv6 time exceeded packet sending (IPv6), 140
 enabling IPv6, 126
 enabling local ND proxy, 137
 enabling local proxy ARP in Layer 3 Ethernet interface view, 15
 enabling local proxy ARP in VLAN interface view, 15
 enabling ND proxy, 136, 137
 enabling Option 82 handling (DHCP), 48
 enabling proxy ARP in Layer 3 Ethernet interface view, 14
 enabling proxy ARP in VLAN interface view, 14
 enabling reception of broadcasts by network (IP performance optimization), 103
 enabling reception of directed broadcast to network (IP performance optimization), 103
 enabling relay agent address check (DHCP), 59
 enabling relay agent offline detection (DHCP), 61
 enabling relay agent starvation attack protection (DHCP), 60
 enabling relay agent unauthorized server detection (DHCP), 60
 enabling replying to multicast echo requests (IPv6), 140
 enabling server on interface (DHCP), 46
 enabling snooping (DHCPv6), 169
 enabling snooping starvation attack protection (DHCP), 77
 enabling support for ICMP extensions, 108
 enabling unauthorized server detection (DHCP), 47
 ensuring client obtains address from authorized server (DHCPv6), 168
 maintaining ARP, 7
 maintaining ARP snooping, 20
 maintaining DHCP server, 49
 maintaining DHCPv6 client, 165
 maintaining DHCPv6 relay agent, 163
 maintaining DHCPv6 server, 157
 maintaining DHCPv6 snooping, 170
 maintaining IP performance optimization, 109
 maintaining IPv4 DNS, 89
 maintaining IPv6 basic configuration, 141
 maintaining IPv6 DNS, 96
 maintaining proxy ARP, 15
 maintaining relay agent (DHCP), 64
 maintaining snooping (DHCP), 78

- maintaining tunneling configuration, 217
- maintaining UDP Helper, 116
- recording client IP-to-MAC mapping (DHCPv6), 169
- setting age timer for ND entries, 134
- setting dynamic entry aging time (ARP), 5
- specifying
 - interface EUI-64 IPv6 address manually (IPv6), 127
- specifying server IP address DHCP client, 44
- specifying trap message send threshold (DHCP), 49

process

- ARP address resolution, 2
- dynamic domain name resolution (DNS), 83
- dynamic IP address allocation (DHCP), 28
- GRE de-encapsulation, 178
- GRE encapsulation, 177

property (TCP), 139

protocols and standards

- BOOTP, 81
- DHCP, 34
- GRE tunneling, 178
- IPv6 NDP, 125
- IRDP, 111
- neighbor discovery (IPv6), 121
- stateless DHCPv6, 152

proxy

- advertised IP addresses, 111
- configuring DNS proxy, 88
- DNS, 84
- DNS operation, 85
- enabling local ND proxy, 137
- enabling ND proxy, 136, 137
- IPv4 DNS proxy configuration, 93

proxy ARP

- common, 13
- configuration, 13, 15
- displaying, 15
- enabling, 14
- enabling local proxy ARP, 14
- isolate-user-VLAN local proxy configuration, 18
- local, 14
- maintaining, 15
- port isolation proxy configuration (local proxy ARP), 16
- super VLAN local proxy configuration, 17

QoS support (IPv6), 118

quick update (ARP), 6

RA destination address, 111

recording client IP-to-MAC mapping (DHCPv6), 169

redirection (IPv6 NDP), 121, 123

relay agent

- application environment (DHCP), 55
- application environment (DHCPv6), 161
- configuration (DHCP), 55, 64
- configuring (DHCPv6), 161, 162, 163
- configuring dynamic client entry periodic refresh (DHCP), 59
- configuring Option 82 support (DHCP), 65
- configuring release of IP address (DHCP), 62
- configuring security function (DHCP), 59
- configuring support for Option 82 (DHCP), 62
- correlating server group with interface (DHCP), 58
- creating static binding (DHCP), 59
- enabling address check (DHCP), 59
- enabling DHCP, 57
- enabling offline detection (DHCP), 61
- enabling relay agent on interface (DHCP), 58
- enabling starvation attack protection (DHCP), 60
- enabling unauthorized server detection (DHCP), 60
- operation (DHCPv6), 161

- Option 82 (DHCP), 32
- support for Option 82 (DHCP), 56
- troubleshooting configuration (DHCP), 66
- UDP Helper configuration, 115, 116
- router/prefix discovery (IPv6 NDP), 121, 123
- routing
 - ARP address resolution process, 2
 - ARP configuration, 1, 4, 7
 - common proxy ARP, 13
 - enabling local proxy ARP, 15
 - enabling proxy ARP, 14
 - gratuitous ARP configuration, 11, 12
 - GRE tunneling, 177
 - IP addressing classes, 21
 - IP addressing configuration, 21, 23
 - IRDP advertising interval, 111
 - local proxy ARP, 14
 - masking (IP addressing), 22
 - multicast ARP configuration, 9
 - PMTU discovery (IPv6), 124
 - proxy ARP configuration, 13, 15
 - redirection (IPv6 NDP), 123
 - special IP addresses, 22
 - subnetting (IP addressing), 22
- security
 - configuring security function (DHCP), 47
 - enabling unauthorized server detection (DHCP), 47
 - IPv6, 118
- selecting address pool (DHCP), 36
- self-defined option
 - configuring (DHCP), 45, 53
 - DHCP, 31
- send/receive buffer (TCP), 105
- server
 - application environment (DHCPv6), 153
 - configuring (DHCPv6), 153, 157
 - configuring address pool (DHCP), 37
 - configuring client BIMS server information (DHCP), 42
 - configuring client DNS server (DHCP), 41
 - configuring client gateway (DHCP), 43
 - configuring client NetBIOS node type (DHCP), 41
 - configuring client TFTP server (DHCP), 44
 - configuring client WINS server (DHCP), 41
 - configuring dynamic IP address assignment (DHCP), 51
 - configuring IP address conflict detection (DHCP), 48
 - configuring security function (DHCP), 47
 - configuring self-defined option (DHCP), 53
 - configuring static IP address assignment (DHCP), 50
 - configuring voice service client Option 184 parameter (DHCP), 43
 - creating address pool (DHCP), 38
 - DHCP configuration, 35, 50
 - DHCPv6 PD, 154
 - DHCPv6 prefix selection process, 154
 - DNS proxy, 84
 - DUID (DHCPv6), 153
 - enabling DHCPv6 server, 155
 - enabling on interface (DHCP), 46
 - enabling Option 82 handling (DHCP), 48
 - enabling relay agent offline detection (DHCP), 61
 - enabling relay agent starvation attack protection (DHCP), 60
 - enabling relay agent unauthorized server detection (DHCP), 60
 - enabling snooping starvation attack protection (DHCP), 77
 - enabling unauthorized server detection (DHCP), 47

- ensuring client obtains address from authorized server (DHCPv6), 168
- ensuring client obtains authorized IP address from server (DHCP), 70
- specifying IP address for DHCP client, 44
- specifying trap message send threshold (DHCP), 49
- troubleshooting DHCP server configuration, 54
- setting
 - age timer for ND entries, 134
 - dynamic entry aging time (ARP), 5
- site-local address (IPv6 unicast), 119
- snooping
 - ARP configuration, 20
 - ARP operation, 20
 - configuring (IPv6 NDP), 135, 136
 - configuring basic functions (DHCP), 74
 - configuring entries backup (DHCP), 76
 - configuring interface max number entries learned (DHCPv6), 170
 - configuring Option 82 support (DHCP), 80
 - configuring support for Option 82 (DHCP), 75
 - configuring trusted port (DHCPv6), 169
 - DHCP configuration, 70, 79
 - DHCPv6 configuration, 168, 170
 - enabling (DHCPv6), 169
 - enabling ARP snooping on VLAN, 20
 - enabling DHCP-REQUEST message attack protection (DHCP), 77
 - enabling starvation attack protection (DHCP), 77
 - ensuring client obtains authorized IP address from server (DHCP), 70
 - function (DHCP), 70
 - recording client IP-to-MAC mapping (DHCP), 70
 - support for Option 82 (DHCP), 72
 - trusted port application environment (DHCP), 71
- special IP addresses, 22
- specifying
 - interface EUI-64 IPv6 address manually (IPv6), 127
 - trap message send threshold (DHCP), 49
- spoofing (DNS), 85, 88
- starvation attack protection
 - relay agent (DHCP), 60
 - snooping (DHCP), 77
- stateless
 - address autoconfiguration (IPv6), 127
 - DHCPv6, 151
 - operation (DHCPv6), 151
- static
 - ARP table entry, 3, 4
 - domain name resolution (DNS), 83
 - domain name resolution (IPv6 DNS), 95
 - neighbor entry (IPv6 NDP), 130
- structure (DHCP address pool), 35
- subnet
 - common proxy ARP, 13
 - enabling local proxy ARP, 15
 - enabling proxy ARP, 14
 - IP addressing, 22
 - local proxy ARP, 14
- subscription service, 219
- suffix
 - DHCP client domain name, 41
 - DNS client list, 84
- support and other resources, 219
- switch (ICMP extension support), 107
- symbols, 220
- table
 - ARP, 3
 - ARP dynamic entry, 3, 4
 - ARP static entry, 3, 4
- TCP

- attribute configuration (IP performance optimization), 105
- configuring property (IPv6), 139
- configuring send/receive buffer size, 105
- configuring timer, 105
- terminology (IRDP), 111
- TFTP (DHCP configuration), 44
- time exceeded packet (IPv6), 140
- timer
 - configuring dynamic PMTU aging timer (IPv6), 138
 - configuring TCP timers, 105
 - setting age timer for ND entries, 134
- transition technologies (IPv6), 124
- Transmission Control Protocol. *See* TCP
- trapping (DHCP), 49
- troubleshooting
 - DHCP server configuration, 54
 - IPv4 DNS configuration, 94
 - IPv6 basic configuration, 148
 - relay agent configuration (DHCP), 66
 - tunneling configuration, 218
- tunneling
 - configuration, 172
 - configuring 6to4 tunnel, 185, 187
 - configuring GRE over IPv4 tunnel, 207, 209, 213
 - configuring GRE over IPv6 tunnel, 212
 - configuring GRE tunnel interface, 179
 - configuring IPv4 over IPv4 tunnel, 194, 195
 - configuring IPv4 over IPv6 tunnel, 198, 199
 - configuring IPv6 manual tunnel, 180
 - configuring IPv6 over IPv6 tunnel, 203, 204
 - displaying configuration, 217
 - GRE, 177
 - GRE de-encapsulation process, 178
 - GRE encapsulation process, 177
 - GRE packet encapsulation format, 177
 - IPv4 over IPv4, 175
 - IPv4 over IPv4 decapsulation, 175
 - IPv4 over IPv4 encapsulation, 175
 - IPv4/IPv6 over IPv6, 176
 - IPv4/IPv6 over IPv6 decapsulation, 176
 - IPv4/IPv6 over IPv6 encapsulation, 176
 - IPv4/IPv6 tunnels, 172
 - IPv6 over IPv4, 173
 - IPv6 over IPv4 automatic, 173
 - IPv6 over IPv4 manually configured, 173
 - ISATAP configuration, 189, 191
 - maintaining configuration, 217
 - troubleshooting configuration, 218
- two message rapid assignment (DHCPv6), 149
- type
 - address (IPv6), 119
 - anycast address (IPv6), 119
 - IPv4 tunneling, 174
 - IPv6 tunneling, 174
 - multicast address (IPv6), 119, 120
 - unicast address (IPv6), 119
- UDP Helper
 - configuration, 115, 116
 - displaying, 116
 - maintaining, 116
- unicast
 - address (IPv6), 119
 - configuring global address (IPv6), 126
 - UDP Helper configuration, 115, 116
- unnumbered IP addressing, 25
- unspecified address (IPv6 unicast), 119
- vendor-specific Option 43 (DHCP), 31
- VLAN
 - ARP snooping configuration), 20

BOOTP client configuration, 81, 82
common proxy ARP, 13
DHCP overview, 27
DHCP server configuration, 35
dynamically obtain address (BOOTP), 81
enabling ARP snooping on VLAN, 20
enabling local proxy ARP in interface view, 15
enabling proxy ARP in interface view, 14
IP performance optimization configuration, 103, 104
IPv6 basic configuration, 117, 126, 143
isolate-user local proxy ARP configuration, 18
local proxy ARP, 14
super VLAN local proxy ARP configuration, 17
TCP attribute configuration (IP performance optimization), 105
UDP Helper configuration, 115, 116
VPN tunneling configuration, 172
websites, 219
Windows Server (multicast ARP for NLB), 6
WINS (DHCP configuration), 41