HP FlexFabric 5700 Switch Series

Network Management and Monitoring Configuration Guide

Part number: 5998-6692

Software version: Release 2416

Document version: 6W100-20150130

Legal and notice information

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

Using ping, tracert, and system debugging

This chapter covers ping, tracert, and information about debugging the system.

Ping

Use the ping utility to determine if a specific address is reachable.

Ping sends ICMP echo requests (ECHO-REQUEST) to the destination device. Upon receiving the requests, the destination device responds with ICMP echo replies (ECHO-REPLY) to the source device. The source device outputs statistics about the ping operation, including the number of packets sent, number of echo replies received, and the round-trip time. You can measure the network performance by analyzing these statistics.

Using a ping command to test network connectivity

Execute **ping** commands in any view.

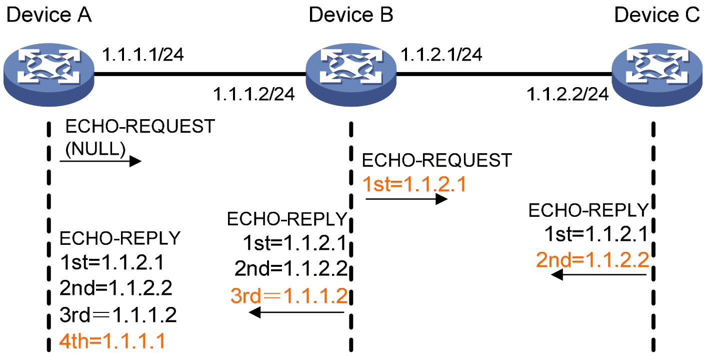
Task	Command
Determine if an address in an IP network is reachable.	When you configure the **ping** command for a low-speed network, set a larger value for the timeout timer (indicated by the **-t** keyword in the command). • For IPv4 networks: **ping** [**ip**] [**-a** *source-ip* | **-c** *count* | **-f** | **-h** *ttl* | **-i** *interface-type interface-number* | **-m** *interval* | **-n** | **-p** *pad* | **-q** | **-r** | **-s** *packet-size* | **-t** *timeout* | **-tos** *tos* | **-v**] * *host* • For IPv6 networks: **ping ipv6** [**-a** *source-ipv6* | **-c** *count* | **-i** *interface-type interface-number* | **-m** *interval* | **-q** | **-s** *packet-size* | **-t** *timeout* | **-v** | **-tc** *traffic-class*] * *host*

Ping example

Network requirements

As shown in Figure 1, determine if Device A and Device C can reach each other. If they can reach each other, get detailed information about routes from Device A to Device C.

Figure 1 Network diagram



Configuration procedure

# Use the **ping** command on Device A to test connectivity to Device C.

```
<DeviceA> ping 1.1.2.2
Ping 1.1.2.2 (1.1.2.2): 56 data bytes, press CTRL_C to break
56 bytes from 1.1.2.2: icmp_seq=0 ttl=254 time=2.137 ms
56 bytes from 1.1.2.2: icmp_seq=1 ttl=254 time=2.051 ms
56 bytes from 1.1.2.2: icmp_seq=2 ttl=254 time=1.996 ms
56 bytes from 1.1.2.2: icmp_seq=3 ttl=254 time=1.963 ms
56 bytes from 1.1.2.2: icmp_seq=4 ttl=254 time=1.991 ms

--- Ping statistics for 1.1.2.2 ---
5 packet(s) transmitted, 5 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 1.963/2.028/2.137/0.062 ms
```

The output shows the following information:

- Device A sends five ICMP packets to Device C and Device A receives five ICMP packets.
- No ICMP packet is lost.
- The route is reachable.

# Get detailed information about routes from Device A to Device C.

```
<DeviceA> ping -r 1.1.2.2
Ping 1.1.2.2 (1.1.2.2): 56 data bytes, press CTRL_C to break
56 bytes from 1.1.2.2: icmp_seq=0 ttl=254 time=4.685 ms
RR:      1.1.2.1
         1.1.2.2
         1.1.1.2
         1.1.1.1
56 bytes from 1.1.2.2: icmp_seq=1 ttl=254 time=4.834 ms  (same route)
56 bytes from 1.1.2.2: icmp_seq=2 ttl=254 time=4.770 ms  (same route)
56 bytes from 1.1.2.2: icmp_seq=3 ttl=254 time=4.812 ms  (same route)
56 bytes from 1.1.2.2: icmp_seq=4 ttl=254 time=4.704 ms  (same route)

--- Ping statistics for 1.1.2.2 ---
5 packet(s) transmitted, 5 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 4.685/4.761/4.834/0.058 ms
```

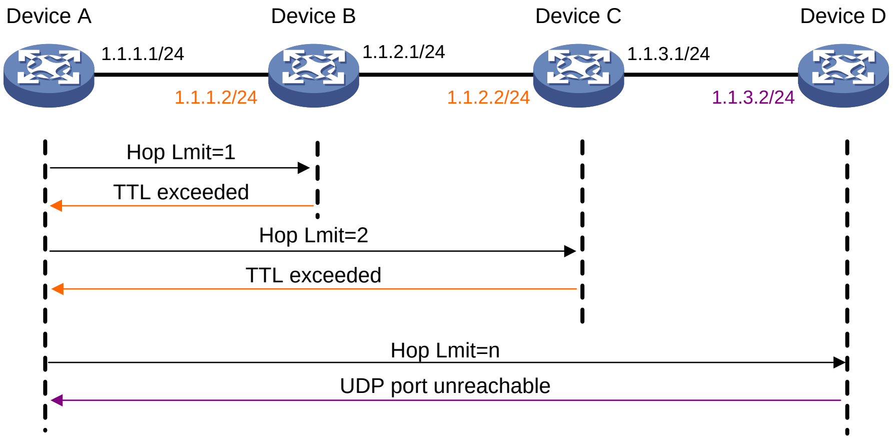
The test procedure of ping –r is as shown in Figure 1:

1. The source device (Device A) sends an ICMP echo request to the destination device (Device C) with the RR option blank.
2. The intermediate device (Device B) adds the IP address of its outbound interface (1.1.2.1) to the RR option of the ICMP echo request, and forwards the packet.
3. Upon receiving the request, the destination device copies the RR option in the request and adds the IP address of its outbound interface (1.1.2.2) to the RR option. Then the destination device sends an ICMP echo reply.
4. The intermediate device adds the IP address of its outbound interface (1.1.1.2) to the RR option in the ICMP echo reply, and then forwards the reply.
5. Upon receiving the reply, the source device adds the IP address of its inbound interface (1.1.1.1) to the RR option. The detailed information of routes between Device A and Device C is formatted as: 1.1.1.1 <-> {1.1.1.2; 1.1.2.1} <-> 1.1.2.2.

Tracert

Tracert (also called Traceroute) retrieves the IP addresses of Layer 3 devices in the path to a specific destination. In the event of network failure, use tracert to test network connectivity and identify failed nodes.

Figure 2 Tracert operation



Tracert uses received ICMP error messages to get the IP addresses of devices. Tracert works as shown in Figure 2:

1. The source device sends a UDP packet with a TTL value of 1 to the destination device. The destination UDP port is not used by any application on the destination device.
2. The first hop (Device B, the first Layer 3 device that receives the packet) responds by sending a TTL-expired ICMP error message to the source, with its IP address (1.1.1.2) encapsulated. This way, the source device can get the address of the first Layer 3 device (1.1.1.2).
3. The source device sends a packet with a TTL value of 2 to the destination device.
4. The second hop (Device C) responds with a TTL-expired ICMP error message, which gives the source device the address of the second Layer 3 device (1.1.2.2).
5. This process continues until a packet sent by the source device reaches the ultimate destination device. Because no application uses the destination port specified in the packet, the destination device responds with a port-unreachable ICMP message to the source device, with its IP address encapsulated. This way, the source device gets the IP address of the destination device (1.1.3.2).

6. The source device thinks that the packet has reached the destination device after receiving the port-unreachable ICMP message, and the path to the destination device is 1.1.1.2 to 1.1.2.2 to 1.1.3.2.

Prerequisites

Before you use a tracert command, perform the tasks in this section.

For an IPv4 network:

- Enable sending of ICMP timeout packets on the intermediate devices (devices between the source and destination devices). If the intermediate devices are HP devices, execute the **ip ttl-expires enable** command on the devices. For more information about this command, see *Layer 3—IP Services Command Reference*.
- Enable sending of ICMP destination unreachable packets on the destination device. If the destination device is an HP device, execute the **ip unreachables enable** command. For more information about this command, see *Layer 3—IP Services Command Reference*.

For an IPv6 network:

- Enable sending of ICMPv6 timeout packets on the intermediate devices (devices between the source and destination devices). If the intermediate devices are HP devices, execute the **ipv6 hoplimit-expires enable** command on the devices. For more information about this command, see *Layer 3—IP Services Command Reference*.
- Enable sending of ICMPv6 destination unreachable packets on the destination device. If the destination device is an HP device, execute the **ipv6 unreachables enable** command. For more information about this command, see *Layer 3—IP Services Command Reference*.

Using a tracert command to identify failed or all nodes in a path

Execute **tracert** commands in any view.

Task	Command
Display the routes from source to destination.	• For IPv4 networks: **tracert** [**-a** *source-ip* | **-f** *first-ttl* | **-m** *max-ttl* | **-p** *port* | **-q** *packet-number* | **-t** *tos* | **-w** *timeout*] * *host* • For IPv6 networks: **tracert ipv6** [**-f** *first-hop* | **-m** *max-hops* | **-p** *port* | **-q** *packet-number* | **-t** *traffic-class* | **-w** *timeout*] * *host*

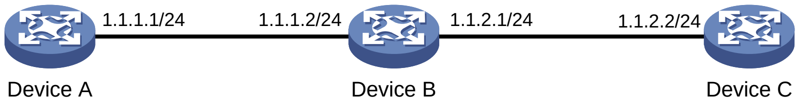
Tracert example

Network requirements

As shown in Figure 3, Device A failed to Telnet to Device C.

Test the network connectivity between Device A and Device C. If they cannot reach each other, locate the failed nodes in the network.

Figure 3 Network diagram



Configuration procedure

1. Configure the IP addresses for devices as shown in Figure 3. (Details not shown.)
2. Configure a static route on Device A.

```
<DeviceA> system-view
[DeviceA] ip route-static 0.0.0.0 0.0.0.0 1.1.1.2
[DeviceA] quit
```

3. Use the **ping** command to test connectivity between Device A and Device C.

```
<DeviceA> ping 1.1.2.2
Ping 1.1.2.2(1.1.2.2): 56 -data bytes, press CTRL_C to break
Request time out
Request time out
Request time out
Request time out
Request time out

--- Ping statistics for 1.1.2.2 ---
5 packet(s) transmitted,0 packet(s) received,100.0% packet loss
```

The output shows that Device A and Device C cannot reach each other.

4. Use the **tracert** command to identify failed nodes:

# Enable sending of ICMP timeout packets on Device B.

```
<DeviceB> system-view
[DeviceB] ip ttl-expires enable
```

# Enable sending of ICMP destination unreachable packets on Device C.

```
<DeviceC> system-view
[DeviceC] ip unreachables enable
```

# Execute the **tracert** command on Device A.

```
<DeviceA> tracert 1.1.2.2
traceroute to 1.1.2.2(1.1.2.2) 30 hops at most,40 bytes each packet, press CTRL_C to
break
 1  1.1.1.2 (1.1.1.2) 1 ms 2 ms 1 ms
 2  * * *
 3  * * *
 4  * * *
 5
<DeviceA>
```

The output shows that Device A can reach Device B but cannot reach Device C. An error has occurred on the connection between Device B and Device C.

5. Use the **debugging ip icmp** command on Device A and Device C to verify that they can send and receive the specific ICMP packets.

Or use the **display ip routing-table** command to verify that there is a route from Device A to Device C.

System debugging

The device supports debugging for the majority of protocols and features and provides debugging information to help users diagnose errors.

Debugging information control switches

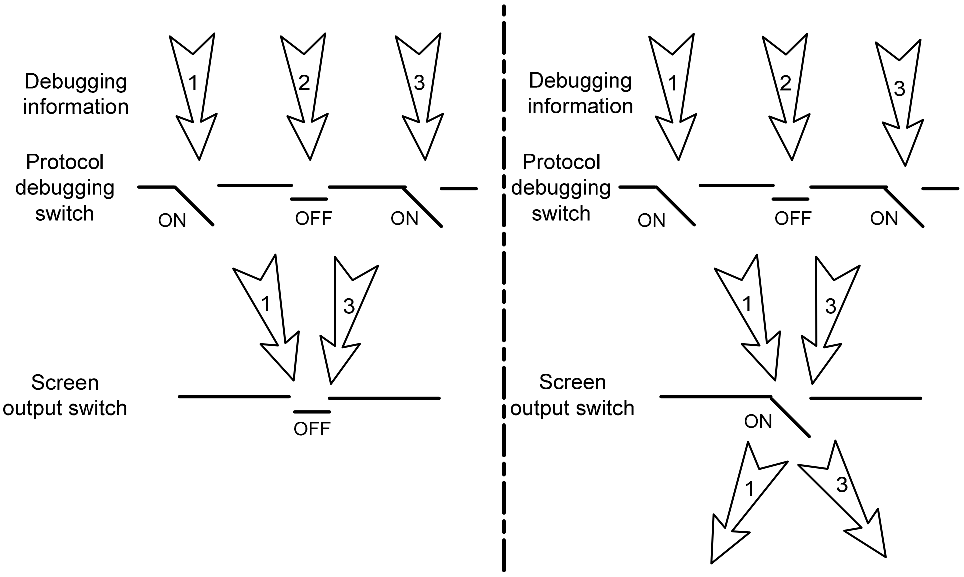
The following switches control the display of debugging information:

- **Module debugging switch**—Controls whether to generate the module-specific debugging information.
- **Screen output switch**—Controls whether to display the debugging information on a certain screen. Use **terminal monitor** and **terminal logging level** commands to turn on the screen output switch. For more information about these two commands, see *Network Management and Monitoring Command Reference*.

As shown in Figure 4, assume that the device can provide debugging for the three modules 1, 2, and 3. The debugging information can be output on a terminal only when both the module debugging switch and the screen output switch are turned on.

Debugging information is typically displayed on a console. You can also send debugging information to other destinations. For more information, see "Configuring the information center"

Figure 4 Relationship between the module and screen output switch



Debugging a feature module

Output of debugging commands is memory intensive. To guarantee system performance, enable debugging only for modules that are in an exceptional condition. When debugging is complete, use the **undo debugging all** command to disable all the debugging functions.

To debug a feature module:

Step	Command	Remarks
1. Enable debugging for a module in user view.	**debugging** { **all** [**timeout** *time*] | *module-name* [*option*] }	By default, all debugging functions are disabled.
2. (Optional.) Display the enabled debugging in any view.	**display debugging** [*module-name*]	N/A

Configuring NTP

Synchronize your device with a trusted time source by using the Network Time Protocol (NTP) or changing the system time before you run it on a live network. Various tasks, including network management, charging, auditing, and distributed computing depend on an accurate system time setting, because the timestamps of system messages and logs use the system time.

Overview

NTP is typically used in large networks to dynamically synchronize time among network devices. It guarantees higher clock accuracy than manual system clock setting. In a small network that does not require high clock accuracy, you can keep time synchronized among devices by changing their system clocks one by one.

NTP runs over UDP and uses UDP port 123.

NOTE:

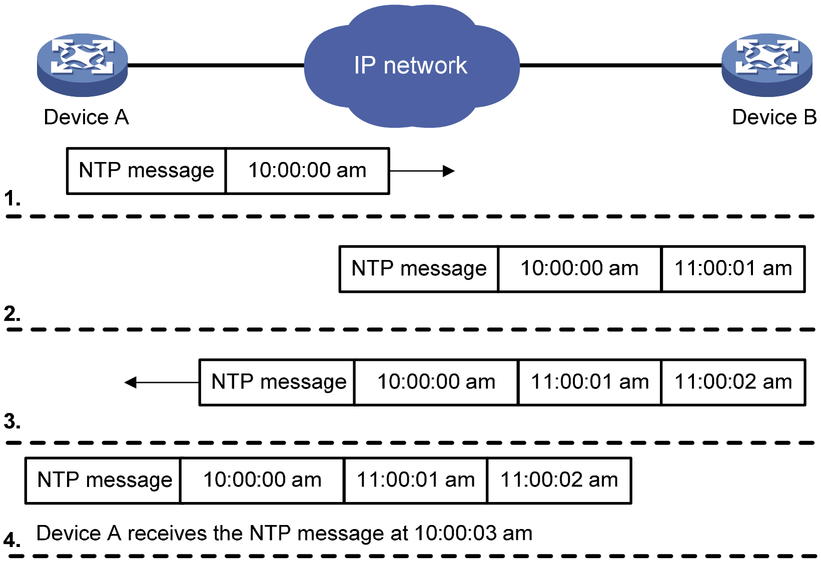
The term "interface" in this chapter collectively refers to Layer 3 interfaces.

How NTP works

Figure 5 shows how NTP synchronizes the system time between two devices, in this example, Device A and Device B. Assume that:

- Prior to the time synchronization, the time is set to 10:00:00 am for Device A and 11:00:00 am for Device B.
- Device B is used as the NTP server. Device A is to be synchronized to Device B.
- It takes 1 second for an NTP message to travel from Device A to Device B, and from Device B to Device A.
- It takes 1 second for Device B to process the NTP message.

Figure 5 Basic work flow



The synchronization process is as follows:

1. Device A sends Device B an NTP message, which is timestamped when it leaves Device A. The time stamp is 10:00:00 am (T1).
2. When this NTP message arrives at Device B, Device B adds a timestamp showing the time when the message arrived at Device B. The timestamp is 11:00:01 am (T2).
3. When the NTP message leaves Device B, Device B adds a timestamp showing the time when the message left Device B. The timestamp is 11:00:02 am (T3).
4. When Device A receives the NTP message, the local time of Device A is 10:00:03 am (T4).

Up to now, Device A can calculate the following parameters based on the timestamps:

- The roundtrip delay of the NTP message: Delay = (T4 – T1) – (T3 – T2) = 2 seconds.
- Time difference between Device A and Device B: Offset = ((T2 – T1) + (T3 – T4)) /2 = 1 hour.

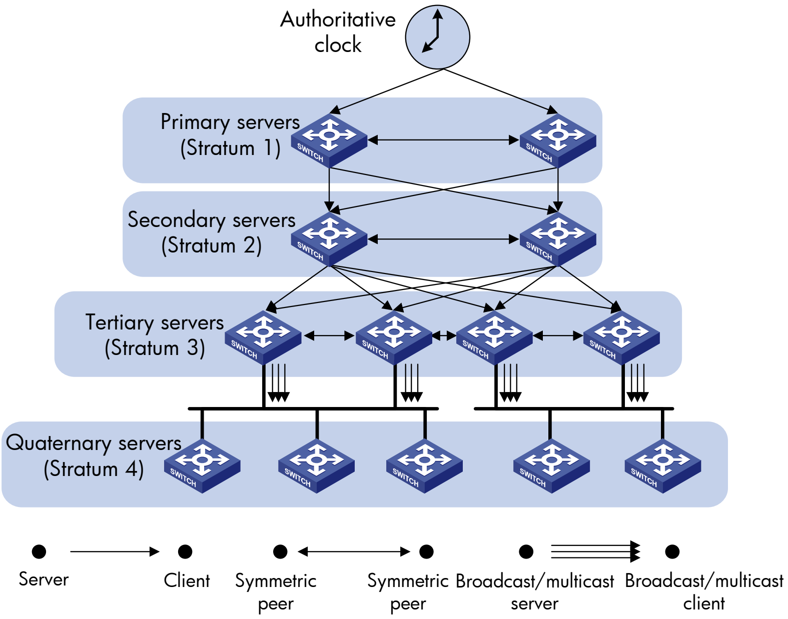
Based on these parameters, Device A can be synchronized to Device B.

This is only a rough description of the work mechanism of NTP. For more information, see the related protocols and standards.

NTP architecture

NTP uses stratums 1 to 16 to define clock accuracy, as shown in Figure 6. A lower stratum value represents higher accuracy. Clocks at stratums 1 through 15 are in synchronized state, and clocks at stratum 16 are not synchronized.

Figure 6 NTP architecture


Authoritative clock
Primary servers (Stratum 1)
Secondary servers (Stratum 2)
Tertiary servers (Stratum 3)
Quaternary servers (Stratum 4)
SWITCH
Server
Client
Symmetric peer
Symmetric peer
Broadcast/multicast server
Broadcast/multicast client

Typically, a stratum 1 NTP server gets its time from an authoritative time source, such as an atomic clock, and provides time for other devices as the primary NTP server. The accuracy of each server is the stratum, with the topmost level (primary servers) assigned as one and each level downwards in the hierarchy assigned as one greater than the preceding level. NTP uses a stratum to describe how many NTP hops away a device is from the primary time server. A stratum 2 time server receives its time from a stratum 1 time server, and so on.

To ensure time accuracy and availability, you can specify multiple NTP servers for a device. The device selects an optimal NTP server as the clock source based on parameters such as stratum. The clock that the device selects is called the reference source. For more information about clock selection, see the related protocols and standards.

If the devices in a network cannot synchronize to an authoritative time source, you can select a device that has a relatively accurate clock from the network, and use the local clock of the device as the reference clock to synchronize other devices in the network.

Association modes

NTP supports the following association modes:

- Client/server mode
- Symmetric active/passive mode
- Broadcast mode
- Multicast mode

Table 1 NTP association modes

Mode	Working process	Principle	Application scenario
Client/server	On the client, specify the IP address of the NTP server. A client sends a clock synchronization message to the NTP servers. Upon receiving the message, the servers automatically operate in server mode and send a reply. If the client can be synchronized to multiple time servers, it selects an optimal clock and synchronizes its local clock to the optimal reference source after receiving the replies from the servers.	A client can be synchronized to a server, but a server cannot be synchronized to a client.	As Figure 6 shows, this mode is intended for configurations where devices of a higher stratum are synchronized to devices with a lower stratum.
Symmetric active/passive	On the symmetric active peer, specify the IP address of the symmetric passive peer. A symmetric active peer periodically sends clock synchronization messages to a symmetric passive peer. The symmetric passive peer automatically operates in symmetric passive mode and sends a reply. If the symmetric active peer can be synchronized to multiple time servers, it selects an optimal clock and synchronizes its local clock to the optimal reference source after receiving the replies from the servers.	A symmetric active peer and a symmetric passive peer can be synchronized to each other. If both of them are synchronized, the peer with a higher stratum is synchronized to the peer with a lower stratum.	As Figure 6 shows, this mode is most often used between two or more servers with the same stratum to operate as a backup for one another. If a server fails to communicate with all the servers of a higher stratum, the server can be synchronized to the servers of the same stratum.

Mode	Working process	Principle	Application scenario
Broadcast	A server periodically sends clock synchronization messages to the broadcast address 255.255.255.255. Clients listen to the broadcast messages from the servers to synchronize to the server according to the broadcast messages. When a client receives the first broadcast message, the client and the server start to exchange messages to calculate the network delay between them. Then, only the broadcast server sends clock synchronization messages.	A broadcast client can be synchronized to a broadcast server, but a broadcast server cannot be synchronized to a broadcast client.	A broadcast server sends clock synchronization messages to synchronize clients in the same subnet. As Figure 6 shows, broadcast mode is intended for configurations involving one or a few servers and a potentially large client population. The broadcast mode has a lower time accuracy than the client/server and symmetric active/passive modes because only the broadcast servers send clock synchronization messages.
Multicast	A multicast server periodically sends clock synchronization messages to the user-configured multicast address. Clients listen to the multicast messages from servers and synchronize to the server according to the received messages.	A multicast client can be synchronized to a multicast server, but a multicast server cannot be synchronized to a multicast client.	A multicast server can provide time synchronization for clients in the same subnet or in different subnets. The multicast mode has a lower time accuracy than the client/server and symmetric active/passive modes.

In this document, an "NTP server" or a "server" refers to a device that operates as an NTP server in client/server mode. Time servers refer to all the devices that can provide time synchronization, including NTP servers, NTP symmetric peers, broadcast servers, and multicast servers.

NTP security

To improve time synchronization security, NTP provides the access control and authentication functions.

NTP access control

You can control NTP access by using an ACL. The access rights are in the following order, from least restrictive to most restrictive:

- **Peer**—Allows time requests and NTP control queries (such as alarms, authentication status, and time server information) and allows the local device to synchronize itself to a peer device.
- **Server**—Allows time requests and NTP control queries, but does not allow the local device to synchronize itself to a peer device.
- **Synchronization**—Allows only time requests from a system whose address passes the access list criteria.
- **Query**—Allows only NTP control queries from a peer device to the local device.

The device processes an NTP request, as follows:

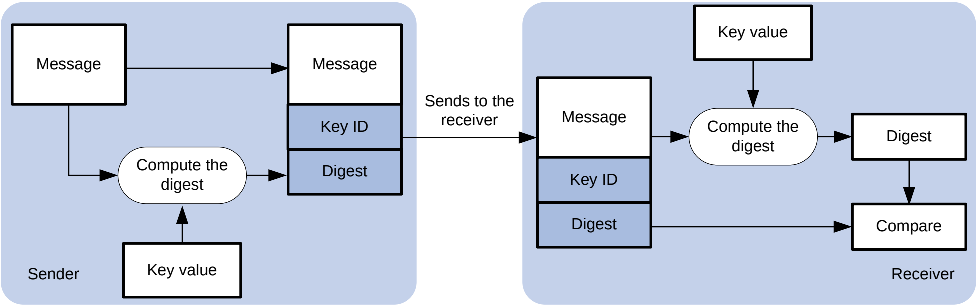
- If no NTP access control is configured, **peer** is granted to the local device and peer devices.
- If the IP address of the peer device matches a **permit** statement in an ACL for more than one access right, the least restrictive access right is granted to the peer device. If a **deny** statement or no ACL is matched, no access right is granted.
- If no ACL is created for an access right, the associated access right is not granted.
- If no ACL is created for any access right, **peer** is granted.

This feature provides minimal security for a system running NTP. A more secure method is NTP authentication.

NTP authentication

Use this feature to authenticate the NTP messages for security purposes. If an NTP message passes authentication, the device can receive it and get time synchronization information. If not, the device discards the message. This function makes sure the device does not synchronize to an unauthorized time server.

Figure 7 NTP authentication



As shown in Figure 7, NTP authentication works as follows:

1. The sender uses the MD5 algorithm to calculate the NTP message according to the key identified by a key ID, and sends the calculated digest together with the NTP message and key ID to the receiver.
2. Upon receiving the message, the receiver finds the key according to the key ID in the message, uses the MD5 algorithm to calculate the digest, and compares the digest with the digest contained in the NTP message. If they are the same, the receiver accepts the message. Otherwise, it discards the message.

Protocols and standards

- RFC 1305, *Network Time Protocol (Version 3) Specification, Implementation and Analysis*
- RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

Configuration restrictions and guidelines

Follow these restrictions and guidelines when you configure NTP:

- You cannot configure both NTP and SNTP on the same device.
- Do not configure NTP on an aggregate member port.
- The NTP service and SNTP service are mutually exclusive. You can only enable either NTP service or SNTP service at a time.

- To ensure time synchronization accuracy, HP recommends not specifying more than one reference source. Doing so might cause frequent time changes or even synchronization failures.
- Make sure you use the **clock protocol** command to specify the time protocol as NTP. For more information about the **clock protocol** command, see *Fundamentals Command Reference*.

Configuration task list

Tasks at a glance
(Required.) Enabling the NTP service
(Required.) Perform one or both of the following tasks: • Configuring NTP association modes • Configuring the local clock as a reference source
(Optional.) Configuring access control rights
(Optional.) Configuring NTP authentication
(Optional.) Configuring NTP optional parameters

Enabling the NTP service

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable the NTP service.	**ntp-service enable**	By default, the NTP service is not enabled.

Configuring NTP association modes

This section describes how to configure NTP association modes.

Configuring NTP in client/server mode

When the device operates in client/server mode, specify the IP address for the server on the client.

Follow these guidelines when you configure an NTP client:

- A server must be synchronized by other devices or use its local clock as a reference source before synchronizing an NTP client. Otherwise, the client will not be synchronized to the NTP server.
- If the stratum level of a server is higher than or equal to a client, the client will not synchronize to that server.
- You can configure multiple servers by repeating the **ntp-service unicast-server** and **ntp-service ipv6 unicast-server** commands.

To configure an NTP client:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Specify an NTP server for the device.	- Specify an NTP server for the device: **ntp-service unicast-server** { *server-name* | *ip-address* } [**authentication-keyid** *keyid* | **priority** | **source** *interface-type interface-number* | **version** *number*] * - Specify an IPv6 NTP server for the device: **ntp-service ipv6 unicast-server** { *server-name* | *ipv6-address* } [**authentication-keyid** *keyid* | **priority** | **source** *interface-type interface-number*] *	By default, no NTP server is specified for the device.

Configuring NTP in symmetric active/passive mode

When the device operates in symmetric active/passive mode, specify on a symmetric-active peer the IP address for a symmetric-passive peer.

Follow these guidelines when you configure a symmetric-active peer:

- Execute the **ntp-service enable** command on a symmetric passive peer to enable NTP. Otherwise, the symmetric-passive peer will not process NTP messages from a symmetric-active peer.
- Either the symmetric-active peer, or the symmetric-passive peer, or both of them must be in synchronized state. Otherwise, their time cannot be synchronized.
- You can configure multiple symmetric-passive peers by repeating the **ntp-service unicast-peer** or **ntp-service ipv6 unicast-peer** command.

To configure a symmetric-active peer:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Specify a symmetric-passive peer for the device.	- Specify a symmetric-passive peer: **ntp-service unicast-peer** { *peer-name* | *ip-address* } [**authentication-keyid** *keyid* | **priority** | **source** *interface-type interface-number* | **version** *number*] * - Specify an IPv6 symmetric-passive peer: **ntp-service ipv6 unicast-peer** { *peer-name* | *ipv6-address* } [**authentication-keyid** *keyid* | **priority** | **source** *interface-type interface-number*] *	By default, no symmetric-passive peer is specified.

Configuring NTP in broadcast mode

A broadcast server must be synchronized by other devices or use its local clock as a reference source before synchronizing a broadcast client. Otherwise, the broadcast client will not be synchronized to the broadcast server.

Configure NTP in broadcast mode on both broadcast server and client.

Configuring a broadcast client

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter interface view.	**interface** *interface-type interface-number*	Enter the interface for receiving NTP broadcast messages.
3.	Configure the device to operate in broadcast client mode.	**ntp-service broadcast-client**	By default, the device does not operate in broadcast client mode. After you execute the command, the device receives NTP broadcast messages from the specified interface.

Configuring the broadcast server

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter interface view.	**interface** *interface-type interface-number*	Enter the interface for sending NTP broadcast messages.
3.	Configure the device to operate in NTP broadcast server mode.	**ntp-service broadcast-server** [**authentication-keyid** *keyid* | **version** *number*] *	By default, the device does not operate in broadcast server mode. After you execute the command, the device receives NTP broadcast messages from the specified interface.

Configuring NTP in multicast mode

A multicast server must be synchronized by other devices or use its local clock as a reference source before synchronizing a multicast client. Otherwise, the multicast client will not be synchronized to the multicast server.

Configure NTP in multicast mode on both a multicast server and client.

Configuring a multicast client

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter interface view.	**interface** *interface-type interface-number*	Enter the interface for receiving NTP multicast messages.

Step	Command	Remarks
3. Configure the device to operate in multicast client mode.	- Configure the device to operate in multicast client mode: **ntp-service multicast-client** [*ip-address*] - Configure the device to operate in IPv6 multicast client mode: **ntp-service ipv6 multicast-client** *ipv6-multicast-address*	By default, the device does not operate in multicast server mode. After you execute the command, the device receives NTP multicast messages from the specified interface.

Configuring the multicast server

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	Enter the interface for sending NTP multicast message.
3. Configure the device to operate in multicast server mode.	- Configure the device to operate in multicast server mode: **ntp-service multicast-server** [*ip-address*] [**authentication-keyid** *keyid* | **ttl** *ttl-number* | **version** *number*] * - Configure the device to operate in multicast server mode: **ntp-service ipv6 multicast-server** *ipv6-multicast-address* [**authentication-keyid** *keyid* | **ttl** *ttl-number*] *	By default, the device does not operate in multicast server mode. After you execute the command, the device receives NTP multicast messages from the specified interface.

Configuring access control rights

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure the NTP service access control right for a peer device to access the local device.	- Configure the NTP service access control right for a peer device to access the local device **ntp-service access** { **peer** | **query** | **server** | **synchronization** } *acl-number* - Configure the IPv6 NTP service access control right for a peer device to access the local device **ntp-service ipv6** { **peer** | **query** | **server** | **synchronization** } **acl** *acl-number*	By default, the NTP service access control right for a peer device to access the local device is peer.

Before you configure the NTP service access control right to the local device, create and configure an ACL associated with the access control right. For more information about ACL, see *ACL and QoS Configuration Guide*.

Configuring NTP authentication

This section provides instructions for configuring NTP authentication.

Configuring NTP authentication in client/server mode

When you configure NTP authentication in client/server mode:

- Enable NTP authentication.
- Configure an authentication key.
- Set the key as a trusted key on both client and server.
- Associate the key with the NTP server on the client.

The key IDs and key values configured on the server and client must be the same. Otherwise, NTP authentication fails.

To configure NTP authentication for a client:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable NTP authentication.	**ntp-service authentication enable**	By default, NTP authentication is disabled.
3. Configure an NTP authentication key.	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no NTP authentication key is configured.
4. Configure the key as a trusted key.	**ntp-service reliable authentication-keyid** *keyid*	By default, no authentication key is configured as a trusted key.
5. Associate the specified key with an NTP server.	• Associate the specified key with an NTP server: **ntp-service unicast-server** { *server-name* | *ip-address* } **authentication-keyid** *keyid* • Associate the specified key with an IPv6 NTP server: **ntp-service ipv6 unicast-server** { *server-name* | *ipv6-address* } **authentication-keyid** *keyid*	N/A

To configure NTP authentication for a server:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable NTP authentication.	**ntp-service authentication enable**	By default, NTP authentication is disabled.

Step		Command	Remarks
3.	Configure an NTP authentication key.	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no NTP authentication key is configured.
4.	Configure the key as a trusted key.	**ntp-service reliable authentication-keyid** *keyid*	By default, no authentication key is configured as a trusted key.

NTP authentication results differ when different configurations are performed on client and server. For more information, see Table 2. (N/A in the table means that whether the configuration is performed does not make any difference.)

Table 2 NTP authentication results

Client			Server		
Enable NTP authentication	**Configure a key and configure it as a trusted key**	**Associate the key with an NTP server**	**Enable NTP authentication**	**Configure a key and configure it as a trusted key**	**Authentication result**
Yes	Yes	Yes	Yes	Yes	Succeeded. NTP messages can be sent and received correctly.
Yes	Yes	Yes	Yes	No	Failed. NTP messages cannot be sent and received correctly.
Yes	Yes	Yes	No	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	No	Yes	N/A	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	N/A	No	N/A	N/A	No authentication. NTP messages can be sent and received correctly.
No	N/A	N/A	N/A	N/A	No authentication. NTP messages can be sent and received correctly.

Configuring NTP authentication in symmetric active/passive mode

When you configure NTP authentication in symmetric peers mode:

- Enable NTP authentication.
- Configure an authentication key.
- Set the key as a trusted key on both active peer and passive peer.
- Associate the key with the passive peer on the active peer.

The key IDs and key values configured on the active peer and passive peer must be the same. Otherwise, NTP authentication fails.

To configure NTP authentication for an active peer:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable NTP authentication.	**ntp-service authentication enable**	By default, NTP authentication is disabled.
3. Configure an NTP authentication key.	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no NTP authentication key is configured.
4. Configure the key as a trusted key.	**ntp-service reliable authentication-keyid** *keyid*	By default, no authentication key is configured as a trusted key.
5. Associate the specified key with a passive peer.	• Associate the specified key with a passive peer: **ntp-service unicast-peer** { *ip-address* | *peer-name* } **authentication-keyid** *keyid* • Associate the specified key with a passive peer: **ntp-service ipv6 unicast-peer** { *ipv6-address* | *peer-name* } **authentication-keyid** *keyid*	N/A

To configure NTP authentication for a passive peer:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable NTP authentication.	**ntp-service authentication enable**	By default, NTP authentication is disabled.
3. Configure an NTP authentication key.	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no NTP authentication key is configured.
4. Configure the key as a trusted key.	**ntp-service reliable authentication-keyid** *keyid*	By default, no authentication key is configured as a trusted key.

NTP authentication results differ when different configurations are performed on active peer and passive peer. For more information, see Table 3. (N/A in the table means that whether the configuration is performed does not make any difference.)

Table 3 NTP authentication results

Active peer			Passive peer		
Enable NTP authentication	**Configure a key and configure it as a trusted key**	**Associate the key with a passive peer**	**Enable NTP authentication**	**Configure a key and configure it as a trusted key**	**Authentication result**
Stratum level of the active and passive peers is not considered.					
Yes	Yes	Yes	Yes	Yes	Succeeded. NTP messages can be sent and received correctly.
Yes	Yes	Yes	Yes	No	Failed. NTP messages cannot be sent and received correctly.
Yes	Yes	Yes	No	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	N/A	No	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	N/A	No	No	N/A	No authentication. NTP messages can be sent and received correctly.
No	N/A	N/A	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
No	N/A	N/A	No	N/A	No authentication. NTP messages can be sent and received correctly.
The active peer has a higher stratum than the passive peer.					
Yes	No	Yes	N/A	N/A	Failed. NTP messages cannot be sent and received correctly.
The passive peer has a higher stratum than the active peer.					
Yes	No	Yes	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	No	Yes	No	N/A	No authentication. NTP messages can be sent and received correctly.

Configuring NTP authentication in broadcast mode

When you configure NTP authentication in broadcast mode:

- Enable NTP authentication.
- Configure an authentication key.
- Set the key as a trusted key on both the broadcast client and server.
- Configure an NTP authentication key on the broadcast server.

The key IDs and key values configured on the broadcast server and client must be the same. Otherwise, NTP authentication fails.

To configure NTP authentication for a broadcast client:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable NTP authentication.	**ntp-service authentication enable**	By default, NTP authentication is disabled.
3. Configure an NTP authentication key.	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no NTP authentication key is configured.
4. Configure the key as a trusted key.	**ntp-service reliable authentication-keyid** *keyid*	By default, no authentication key is configured as a trusted key.

To configure NTP authentication for a broadcast server:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable NTP authentication.	**ntp-service authentication enable**	By default, NTP authentication is disabled.
3. Configure an NTP authentication key.	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no NTP authentication key is configured.
4. Configure the key as a trusted key.	**ntp-service reliable authentication-keyid** *keyid*	By default, no authentication key is configured as a trusted key.
5. Enter interface view.	**interface** *interface-type interface-number*	N/A
6. Associate the specified key with the broadcast server.	**ntp-service broadcast-server authentication-keyid** *keyid*	By default, the broadcast server is not associated with any key.

NTP authentication results differ when different configurations are performed on broadcast client and server. For more information, see Table 4. (N/A in the table means that whether the configuration is performed does not make any difference.)

Table 4 NTP authentication results

Broadcast server			Broadcast client		
Enable NTP authentication	**Configure a key and configure it as a trusted key**	**Associate the key with a broadcast server**	**Enable NTP authentication**	**Configure a key and configure it as a trusted key**	**Authentication result**
Yes	Yes	Yes	Yes	Yes	Succeeded. NTP messages can be sent and received correctly.
Yes	Yes	Yes	Yes	No	Failed. NTP messages cannot be sent and received correctly.
Yes	Yes	Yes	No	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	No	Yes	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	No	Yes	No	N/A	No authentication. NTP messages can be sent and received correctly.
Yes	N/A	No	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	N/A	No	No	N/A	No authentication. NTP messages can be sent and received correctly.
No	N/A	N/A	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
No	N/A	N/A	No	N/A	No authentication. NTP messages can be sent and received correctly.

Configuring NTP authentication in multicast mode

When you configure NTP authentication in multicast mode:

- Enable NTP authentication.
- Configure an authentication key.
- Set the key as a trusted key on both the multicast client and server.
- Configure an NTP authentication key on the multicast server.

The key IDs and key values configured on the multicast server and client must be the same. Otherwise, NTP authentication fails.

To configure NTP authentication for a multicast client:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable NTP authentication.	**ntp-service authentication enable**	By default, NTP authentication is disabled.
3.	Configure an NTP authentication key.	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no NTP authentication key is configured.
4.	Configure the key as a trusted key.	**ntp-service reliable authentication-keyid** *keyid*	By default, no authentication key is configured as a trusted key.

To configure NTP authentication for a multicast server:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable NTP authentication.	**ntp-service authentication enable**	By default, NTP authentication is disabled.
3.	Configure an NTP authentication key.	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no NTP authentication key is configured.
4.	Configure the key as a trusted key.	**ntp-service reliable authentication-keyid** *keyid*	By default, no authentication key is configured as a trusted key.
5.	Enter interface view.	**interface** *interface-type interface-number*	N/A
6.	Associate the specified key with the multicast server.	• Associate the specified key with a multicast server: **ntp-service multicast-server** [*ip-address*] **authentication-keyid** *keyid* • Associate the specified key with an IPv6 multicast server: **ntp-service ipv6 multicast-server** *ipv6-multicast-address* **authentication-keyid** *keyid*	By default, no multicast server is associated with the specified key.

NTP authentication results differ when different configurations are performed on broadcast client and server. For more information, see Table 5. (N/A in the table means that whether the configuration is performed does not make any difference.)

Table 5 NTP authentication results

Multicast server			Multicast client		
Enable NTP authentication	**Configure a key and configure it as a trusted key**	**Associate the key with a multicast server**	**Enable NTP authentication**	**Configure a key and configure it as a trusted key**	**Authentication result**
Yes	Yes	Yes	Yes	Yes	Succeeded. NTP messages can be sent and received correctly.
Yes	Yes	Yes	Yes	No	Failed. NTP messages cannot be sent and received correctly.
Yes	Yes	Yes	No	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	No	Yes	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	No	Yes	No	N/A	No authentication. NTP messages can be sent and received correctly.
Yes	N/A	No	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
Yes	N/A	No	No	N/A	No authentication. NTP messages can be sent and received correctly.
No	N/A	N/A	Yes	N/A	Failed. NTP messages cannot be sent and received correctly.
No	N/A	N/A	No	N/A	No authentication. NTP messages can be sent and received correctly.

Configuring NTP optional parameters

The configuration tasks in this section are optional tasks. Configure them to improve NTP security, performance, or reliability.

Specifying the source interface for NTP messages

To prevent interface status changes from causing NTP communication failures, configure the device to use the IP address of an interface that is always up, for example, a loopback interface, as the source IP address for the NTP messages to be sent. Set the loopback interface as the source interface so that any interface status change on the device will not cause NTP messages to be unable to be received.

When the device responds to an NTP request, the source IP address of the NTP response is always the IP address of the interface that has received the NTP request.

Follow these guidelines when you specify the source interface for NTP messages:

- If you have specified the source interface for NTP messages in the **ntp-service** [**ipv6**] **unicast-server** or **ntp-service** [**ipv6**] **unicast-peer** command, the interface specified in the **ntp-service** [**ipv6**] **unicast-server** or **ntp-service** [**ipv6**] **unicast-peer** command works as the source interface for NTP messages.
- If you have configured the **ntp-service broadcast-server** or **ntp-service** [**ipv6**] **multicast-server** command, the source interface for the broadcast or multicast NTP messages is the interface configured with the respective command.

To specify the source interface for NTP messages:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Specify the source interface for NTP messages.	• Specify the source interface for NTP messages: **ntp-service source** *interface-type interface-number* • Specify the source interface for IPv6 NTP messages: **ntp-service ipv6 source** *interface-type interface-number*	By default, no source interface is specified for NTP messages.

Disabling an interface from processing NTP messages

When NTP is enabled, all interfaces by default can process NTP messages. For security purposes, you can disable some of the interfaces from processing NTP messages.

To disable an interface from processing NTP messages:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Disable the interface from processing NTP messages.	• For IPv4: **undo ntp-service inbound enable** • For IPv6: **undo ntp-service ipv6 inbound enable**	By default, an interface processes NTP messages.

Configuring the maximum number of dynamic associations

NTP has the following types of associations:

- **Static association**—A manually created association.
- **Dynamic association**—Temporary association created by the system during NTP operation. A dynamic association is removed if no messages are exchanged within about 12 minutes.

The following describes how an association is established in different association modes:

- **Client/server mode**—After you specify an NTP server, the system creates a static association on the client. The server simply responds passively upon the receipt of a message, rather than creating an association (static or dynamic).
- **Symmetric active/passive mode**—After you specify a symmetric-passive peer on a symmetric active peer, static associations are created on the symmetric-active peer, and dynamic associations are created on the symmetric-passive peer.
- **Broadcast or multicast mode**—Static associations are created on the server, and dynamic associations are created on the client.

A single device can have a maximum of 128 concurrent associations, including static associations and dynamic associations.

Perform this task to restrict the number of dynamic associations to prevent dynamic associations from occupying too many system resources.

To configure the maximum number of dynamic associations:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Configure the maximum number of dynamic sessions allowed to be established.	**ntp-service max-dynamic-sessions** *number*	By default, the command can establish up to 100 dynamic sessions.

Configuring a DSCP value for NTP packets

The DSCP value determines the sending precedence of a packet.

To configure a DSCP value for NTP packets:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Set a DSCP value for NTP packets.	• IPv4 packets: **ntp-service dscp** *dscp-value* • IPv6 packets: **ntp-service ipv6 dscp** *dscp-value*	The defaults for a DSCP value: • 48 for IPv4 NTP packets. • 56 for IPv6 NTP packets.

Configuring the local clock as a reference source

Follow these guidelines when you configure the local clock as a reference source:

- Make sure the local clock can provide the time accuracy required for the network. After you configure the local clock as a reference source, the local clock is synchronized, and can operate as a time server to synchronize other devices in the network. If the local clock is incorrect, timing errors occur.
- Before you configure this feature, adjust the local system time to make sure it is accurate.

To configure the local clock as a reference source:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure the local clock as a reference source.	**ntp-service refclock-master** [*ip-address*] [*stratum*]	By default, the device does not use the local clock as a reference source.

Displaying and maintaining NTP

Execute **display** commands in any view.

Task	Command
Display information about IPv6 NTP associations.	**display ntp-service ipv6 sessions** [**verbose**]
Display information about IPv4 NTP associations.	**display ntp-service sessions** [**verbose**]
Display information about NTP service status.	**display ntp-service status**
Display brief information about the NTP servers from the local device back to the primary reference source.	**display ntp-service trace**

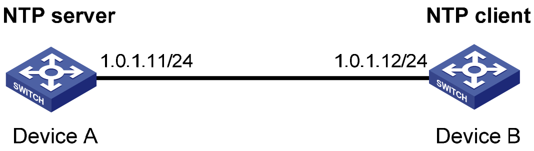
NTP client/server mode configuration example

Network requirements

As shown in Figure 8:

- Configure the local clock of Device A as a reference source, with the stratum level 2.
- Configure Device B to operate in client mode and Device A to be used as the NTP server for Device B.

Figure 8 Network diagram



Configuration procedure

1. Set the IP address for each interface, and make sure Device A and Device B can reach each other, as shown in Figure 8. (Details not shown.)
2. Configure Device A:

# Enable the NTP service.

```
<DeviceA> system-view
[DeviceA] ntp-service enable
```

# Specify the local clock as the reference source, with the stratum level 2.

```
[DeviceA] ntp-service refclock-master 2
```

3. Configure Device B:

# Enable the NTP service.

```
<DeviceB> system-view
[DeviceB] ntp-service enable
```

# Specify Device A as the NTP server of Device B so that Device B is synchronized to Device A.

```
[DeviceB] ntp-service unicast-server 1.0.1.11
```

4. Verify the configuration:

# Verify that Device B has synchronized to Device A, and the clock stratum level is 3 on Device B and 2 on Device A.

```
[DeviceB] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 1.0.1.11
 Local mode: client
 Reference clock ID: 1.0.1.11
 Leap indicator: 00
 Clock jitter: 0.000977 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00383 ms
 Root dispersion: 16.26572 ms
 Reference time: d0c6033f.b9923965  Wed, Dec 29 2010 18:58:07.724
```

# Verify that an IPv4 NTP association has been established between Device B and Device A.

```
[DeviceB] display ntp-service sessions
       source          reference       stra reach poll  now offset  delay disper
********************************************************************************
[12345]1.0.1.11        127.127.1.0        2     1   64   15   -4.0 0.0038 16.262
Notes: 1 source(master), 2 source(peer), 3 selected, 4 candidate, 5 configured.
 Total sessions: 1
```

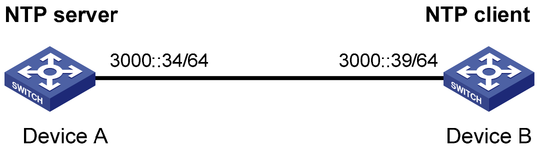
IPv6 NTP client/server mode configuration example

Network requirements

As shown in Figure 9:

- Configure the local clock of Device A as a reference source, with the stratum level 2.
- Configure Device B to operate in client mode and Device A to be used as the IPv6 NTP server for Device B.

Figure 9 Network diagram



Configuration procedure

1. Set the IP address for each interface, and make sure Device A and Device B can reach each other, as shown in Figure 9. (Details not shown.)
2. Configure Device A:

   # Enable the NTP service.

   ```
   <DeviceA> system-view
   [DeviceA] ntp-service enable
   ```

   # Specify the local clock as the reference source, with the stratum level 2.

   ```
   [DeviceA] ntp-service refclock-master 2
   ```

3. Configure Device B:

   # Enable the NTP service.

   ```
   <DeviceB> system-view
   [DeviceB] ntp-service enable
   ```

   # Specify Device A as the IPv6 NTP server of Device B so that Device B is synchronized to Device A.

   ```
   [DeviceB] ntp-service ipv6 unicast-server 3000::34
   ```

4. Verify the configuration:

   # Verify that Device B has synchronized to Device A, and the clock stratum level is 3 on Device B and 2 on Device A.

   ```
   [DeviceB] display ntp-service status
    Clock status: synchronized
    Clock stratum: 3
    System peer: 3000::34
    Local mode: client
    Reference clock ID: 163.29.247.19
    Leap indicator: 00
    Clock jitter: 0.000977 s
    Stability: 0.000 pps
    Clock precision: 2^-10
    Root delay: 0.02649 ms
    Root dispersion: 12.24641 ms
    Reference time: d0c60419.9952fb3e  Wed, Dec 29 2010 19:01:45.598
   ```

   # Verify that an IPv6 NTP association has been established between Device B and Device A.

   ```
   [DeviceB] display ntp-service ipv6 sessions
   Notes: 1 source(master), 2 source(peer), 3 selected, 4 candidate, 5 configured.

    Source: [12345]3000::34
    Reference: 127.127.1.0          Clock stratum: 2
    Reachabilities: 15              Poll interval: 64
   ```

```
 Last receive time: 19          Offset: 0.0
 Roundtrip delay: 0.0           Dispersion: 0.0

 Total sessions: 1
```

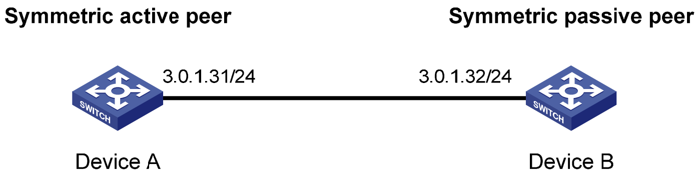
NTP symmetric active/passive mode configuration example

Network requirements

As shown in Figure 10:

- Configure the local clock of Device A as a reference source, with the stratum level 2.
- Configure Device A to operate in symmetric-active mode and specify Device B as the passive peer of Device A.

Figure 10 Network diagram



Configuration procedure

1. Set the IP address for each interface, and make sure Device A and Device B can reach each other, as shown in Figure 10. (Details not shown.)
2. Configure Device B:

   # Enable the NTP service.

   ```
   <DeviceB> system-view
   [DeviceB] ntp-service enable
   ```
3. Configure Device A:

   # Enable the NTP service.

   ```
   <DeviceA> system-view
   [DeviceA] ntp-service enable
   ```

   # Specify the local clock as the reference source, with the stratum level 2.

   ```
   [DeviceA] ntp-service refclock-master 2
   ```

   # Configure Device B as a symmetric passive peer.

   ```
   [DeviceA] ntp-service unicast-peer 3.0.1.32
   ```
4. Verify the configuration:

   # Verify that Device B has synchronized to Device A.

   ```
   [DeviceB] display ntp-service status
    Clock status: synchronized
    Clock stratum: 3
    System peer: 3.0.1.31
    Local mode: sym_passive
    Reference clock ID: 3.0.1.31
   ```

```
 Leap indicator: 00
 Clock jitter: 0.000916 s
 Stability: 0.000 pps
 Clock precision: 2^-17
 Root delay: 0.00609 ms
 Root dispersion: 1.95859 ms
 Reference time: 83aec681.deb6d3e5  Wed, Jan  8 2014 14:33:11.081
```

# Verify that an IPv4 NTP association has been established between Device B and Device A.

```
[DeviceB] display ntp-service sessions
       source          reference       stra reach poll  now offset  delay disper
********************************************************************************
   [12]3.0.1.31        127.127.1.0        2    62   64   34 0.4251 6.0882 1392.1
Notes: 1 source(master), 2 source(peer), 3 selected, 4 candidate, 5 configured.
 Total sessions: 1
```

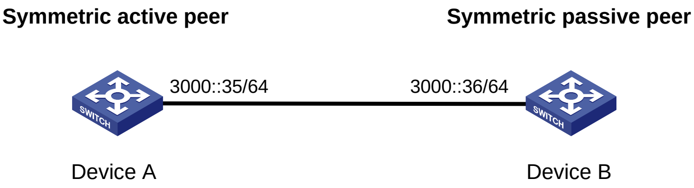
IPv6 NTP symmetric active/passive mode configuration example

Network requirements

As shown in Figure 11:

- Configure the local clock of Device A as a reference source, with the stratum level 2.
- Configure Device A to operate in symmetric-active mode and specify Device B as the IPv6 passive peer of Device A.

Figure 11 Network diagram



Configuration procedure

1. Set the IP address for each interface as shown in Figure 11. (Details not shown.)
2. Configure Device B:

   # Enable the NTP service.

   ```
   <DeviceB> system-view
   [DeviceB] ntp-service enable
   ```
3. Configure Device A:

   # Enable the NTP service.

   ```
   <DeviceA> system-view
   [DeviceA] ntp-service enable
   ```

   # Specify the local clock as the reference source, with the stratum level 2.

   ```
   [DeviceA] ntp-service refclock-master 2
   ```

   # Configure Device B as an IPv6 symmetric passive peer.

```
[DeviceA] ntp-service ipv6 unicast-peer 3000::36
```

4. Verify the configuration:

   # Verify that Device B has synchronized to Device A.

   ```
   [DeviceB] display ntp-service status
    Clock status: synchronized
    Clock stratum: 3
    System peer: 3000::35
    Local mode: sym_passive
    Reference clock ID: 251.73.79.32
    Leap indicator: 11
    Clock jitter: 0.000977 s
    Stability: 0.000 pps
    Clock precision: 2^-10
    Root delay: 0.01855 ms
    Root dispersion: 9.23483 ms
    Reference time: d0c6047c.97199f9f  Wed, Dec 29 2010 19:03:24.590
   ```

   # Verify that an IPv6 NTP association has been established between Device B and Device A.

   ```
   [DeviceB] display ntp-service ipv6 sessions
   Notes: 1 source(master), 2 source(peer), 3 selected, 4 candidate, 5 configured.

    Source: [1234]3000::35
    Reference: 127.127.1.0           Clock stratum: 2
    Reachabilities: 15               Poll interval: 64
    Last receive time: 19            Offset: 0.0
    Roundtrip delay: 0.0             Dispersion: 0.0

    Total sessions: 1
   ```

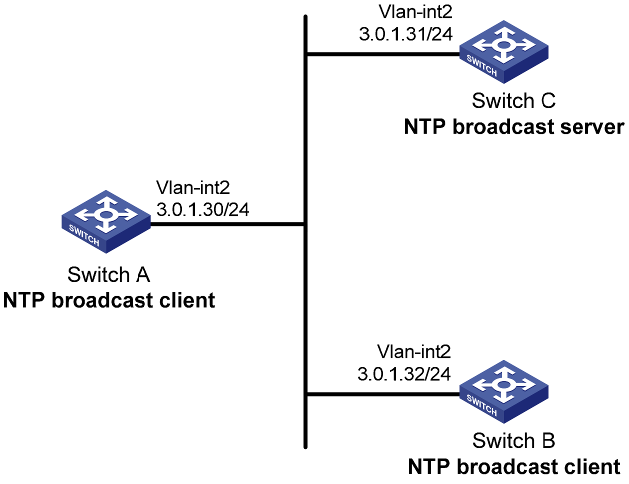
NTP broadcast mode configuration example

Network requirements

As shown in Figure 12, Switch C functions as the NTP server for multiple devices on a network segment and synchronizes the time among multiple devices.

- Configure Switch C's local clock as a reference source, with the stratum level 2.
- Configure Switch C to operate in broadcast server mode and send out broadcast messages from VLAN-interface 2.
- Configure Switch A and Switch B to operate in broadcast client mode, and listen to broadcast messages through VLAN-interface 2.

Figure 12 Network diagram



Configuration procedure

1. Set the IP address for each interface, and make sure Switch A, Switch B, and Switch C can reach each other, as shown in Figure 12. (Details not shown.)
2. Configure Switch C:

   # Enable the NTP service.

   ```
   <SwitchC> system-view
   [SwitchC] ntp-service enable
   ```

   # Specify the local clock as the reference source, with the stratum level 2.

   ```
   [SwitchC] ntp-service refclock-master 2
   ```

   # Configure Switch C to operate in broadcast server mode and send broadcast messages through VLAN-interface 2.

   ```
   [SwitchC] interface vlan-interface 2
   [SwitchC-Vlan-interface2] ntp-service broadcast-server
   ```
3. Configure Switch A:

   # Enable the NTP service.

   ```
   <SwitchA> system-view
   [SwitchA] ntp-service enable
   ```

   # Configure Switch A to operate in broadcast client mode and receive broadcast messages on VLAN-interface 2.

   ```
   [SwitchA] interface vlan-interface 2
   [SwitchA-Vlan-interface2] ntp-service broadcast-client
   ```
4. Configure Switch B:

   # Enable the NTP service.

   ```
   <SwitchB> system-view
   [SwitchB] ntp-service enable
   ```

   # Configure Switch B to operate in broadcast client mode and receive broadcast messages on VLAN-interface 2.

   ```
   [SwitchB] interface vlan-interface 2
   [SwitchB-Vlan-interface2] ntp-service broadcast-client
   ```
5. Verify the configuration:

# Verify that Switch A has synchronized to Switch C, and the clock stratum level is 3 on Switch A and 2 on Switch C.

```
[SwitchA-Vlan-interface2] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3.0.1.31
 Local mode: bclient
 Reference clock ID: 3.0.1.31
 Leap indicator: 00
 Clock jitter: 0.044281 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00229 ms
 Root dispersion: 4.12572 ms
 Reference time: d0d289fe.ec43c720  Sat, Jan  8 2011  7:00:14.922
```

# Verify that an IPv4 NTP association has been established between Switch A and Switch C.

```
[SwitchA-Vlan-interface2] display ntp-service sessions
       source          reference       stra reach poll  now offset  delay disper
********************************************************************************
 [1245]3.0.1.31        127.127.1.0        2     1   64   519    -0.0 0.0022 4.1257
Notes: 1 source(master),2 source(peer),3 selected,4 candidate,5 configured.
 Total sessions: 1
```

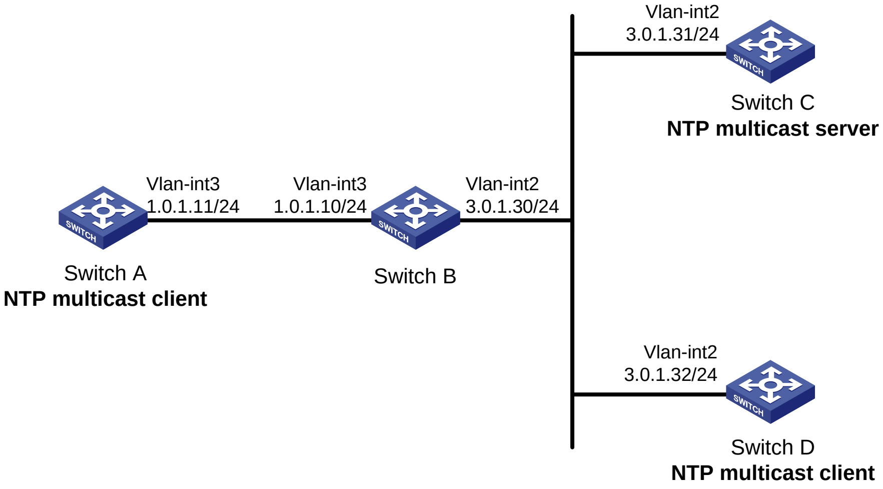
NTP multicast mode configuration example

Network requirements

As shown in Figure 13, Switch C functions as the NTP server for multiple devices on different network segments and synchronizes the time among multiple devices.

- Configure Switch C's local clock as a reference source, with the stratum level 2.
- Configure Switch C to operate in multicast server mode and send out multicast messages from VLAN-interface 2.
- Configure Switch A and Switch D to operate in multicast client mode and receive multicast messages through VLAN-interface 3 and VLAN-interface 2, respectively.

Figure 13 Network diagram



Configuration procedure

In this example, Switch B must support IPv4 multicast routing.

1. Set the IP address for each interface, and make sure the switches can reach each other, as shown in Figure 13. (Details not shown.)
2. Configure Switch C:

   # Enable the NTP service.

   ```
   <SwitchC> system-view
   [SwitchC] ntp-service enable
   ```

   # Specify the local clock as the reference source, with the stratum level 2.

   ```
   [SwitchC] ntp-service refclock-master 2
   ```

   # Configure Switch C to operate in multicast server mode and send multicast messages through VLAN-interface 2.

   ```
   [SwitchC] interface vlan-interface 2
   [SwitchC-Vlan-interface2] ntp-service multicast-server
   ```

3. Configure Switch D:

   # Enable the NTP service.

   ```
   <SwitchD> system-view
   [SwitchD] ntp-service enable
   ```

   # Configure Switch D to operate in multicast client mode and receive multicast messages on VLAN-interface 2.

   ```
   [SwitchD] interface vlan-interface 2
   [SwitchD-Vlan-interface2] ntp-service multicast-client
   ```

4. Verify the configuration:

   Switch D and Switch C are on the same subnet, so Switch D can do the following:

   - Receive the multicast messages from Switch C without being enabled with the multicast functions.
   - Synchronize to Switch C.

   # Verify that Switch D has synchronized to Switch C, and the clock stratum level is 3 on Switch D and 2 on Switch C.

```
[SwitchD-Vlan-interface2] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3.0.1.31
 Local mode: bclient
 Reference clock ID: 3.0.1.31
 Leap indicator: 00
 Clock jitter: 0.044281 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00229 ms
 Root dispersion: 4.12572 ms
 Reference time: d0d289fe.ec43c720  Sat, Jan  8 2011  7:00:14.922
```

# Verify that an IPv4 NTP association has been established between Switch D and Switch C.

```
[SwitchD-Vlan-interface2] display ntp-service sessions
       source          reference       stra reach poll  now offset  delay disper
********************************************************************************

 [1245]3.0.1.31        127.127.1.0        2     1   64  519    -0.0 0.0022 4.1257
Notes: 1 source(master),2 source(peer),3 selected,4 candidate,5 configured.
 Total sessions: 1
```

5. Configure Switch B:

   Because Switch A and Switch C are on different subnets, you must enable the multicast functions on Switch B before Switch A can receive multicast messages from Switch C.

   # Enable IP multicast routing and IGMP.

   ```
   <SwitchB> system-view
   [SwitchB] multicast routing
   [SwitchB-mrib] quit
   [SwitchB] interface vlan-interface 2
   [SwitchB-Vlan-interface2] pim dm
   [SwitchB-Vlan-interface2] quit
   [SwitchB] vlan 3
   [SwitchB-vlan3] port ten-gigabitethernet 1/0/1
   [SwitchB-vlan3] quit
   [SwitchB] interface vlan-interface 3
   [SwitchB-Vlan-interface3] igmp enable
   [SwitchB-Vlan-interface3] igmp static-group 224.0.1.1
   [SwitchB-Vlan-interface3] quit
   [SwitchB] igmp-snooping
   [SwitchB-igmp-snooping] quit
   [SwitchB] interface ten-gigabitethernet 1/0/1
   [SwitchB- Ten-GigabitEthernet1/0/1] igmp-snooping static-group 224.0.1.1 vlan 3
   ```

6. Configure Switch A:

   # Enable the NTP service.

   ```
   <SwitchA> system-view
   [SwitchA] ntp-service enable
   ```

   # Configure Switch A to operate in multicast client mode and receive multicast messages on VLAN-interface 3.

```
[SwitchA] interface vlan-interface 3
[SwitchA-Vlan-interface3] ntp-service multicast-client
```

7. Verify the configuration:

   # Verify that Switch A has synchronized to Switch C, and the clock stratum level is 3 on Switch A and 2 on Switch C.

   ```
   [SwitchA-Vlan-interface3] display ntp-service status
    Clock status: synchronized
    Clock stratum: 3
    System peer: 3.0.1.31
    Local mode: bclient
    Reference clock ID: 3.0.1.31
    Leap indicator: 00
    Clock jitter: 0.165741 s
    Stability: 0.000 pps
    Clock precision: 2^-10
    Root delay: 0.00534 ms
    Root dispersion: 4.51282 ms
    Reference time: d0c61289.10b1193f  Wed, Dec 29 2010 20:03:21.065
   ```

   # Verify that an IPv4 NTP association has been established between Switch A and Switch C.

   ```
   [SwitchA-Vlan-interface3] display ntp-service sessions
          source          reference       stra reach poll  now offset  delay disper
   ********************************************************************************
    [1234]3.0.1.31        127.127.1.0       2   247   64  381   -0.0 0.0053 4.5128
   Notes: 1 source(master),2 source(peer),3 selected,4 candidate,5 configured.
    Total sessions: 1
   ```

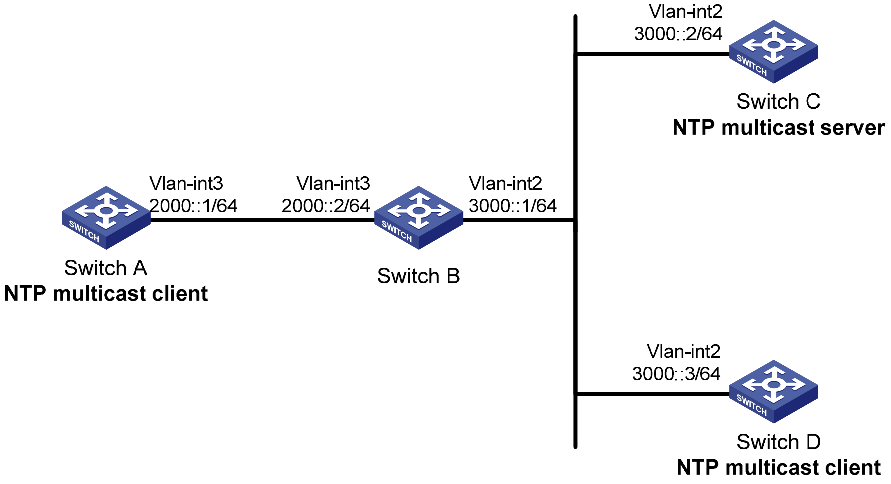
IPv6 NTP multicast mode configuration example

Network requirements

As shown in Figure 14, Switch C functions as the NTP server for multiple devices on different network segments and synchronizes the time among multiple devices.

- Configure Switch C's local clock as a reference source, with the stratum level 2.
- Configure Switch C to operate in IPv6 multicast server mode and send out IPv6 multicast messages from VLAN-interface 2.
- Configure Switch A and Switch D to operate in IPv6 multicast client mode and receive IPv6 multicast messages through VLAN-interface 3 and VLAN-interface 2, respectively.

Figure 14 Network diagram



Configuration procedure

In this example, Switch B must support IPv6 multicast routing.

1. Set the IP address for each interface, and make sure the switches can reach each other, as shown in Figure 14. (Details not shown.)
2. Configure Switch C:

   # Enable the NTP service.

   ```
   <SwitchC> system-view
   [SwitchC] ntp-service enable
   ```

   # Specify the local clock as the reference source, with the stratum level 2.

   ```
   [SwitchC] ntp-service refclock-master 2
   ```

   # Configure Switch C to operate in IPv6 multicast server mode and send multicast messages through VLAN-interface 2.

   ```
   [SwitchC] interface vlan-interface 2
   [SwitchC-Vlan-interface2] ntp-service ipv6 multicast-server ff24::1
   ```
3. Configure Switch D:

   # Enable the NTP service.

   ```
   <SwitchD> system-view
   [SwitchD] ntp-service enable
   ```

   # Configure Switch D to operate in IPv6 multicast client mode and receive multicast messages on VLAN-interface 2.

   ```
   [SwitchD] interface vlan-interface 2
   [SwitchD-Vlan-interface2] ntp-service ipv6 multicast-client ff24::1
   ```
4. Verify the configuration:

   Switch D and Switch C are on the same subnet, so Switch D can do the following:

   - Receive the IPv6 multicast messages from Switch C without being enabled with the IPv6 multicast functions.
   - Synchronize to Switch C.

   # Verify that Switch D has synchronized to Switch C, and the clock stratum level is 3 on Switch D and 2 on Switch C.

   ```
   [SwitchD-Vlan-interface2] display ntp-service status
   ```

```
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3000::2
 Local mode: bclient
 Reference clock ID: 165.84.121.65
 Leap indicator: 00
 Clock jitter: 0.000977 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00000 ms
 Root dispersion: 8.00578 ms
 Reference time: d0c60680.9754fb17  Wed, Dec 29 2010 19:12:00.591
```

# Verify that an IPv6 NTP association has been established between Switch D and Switch C.

```
[SwitchD-Vlan-interface2] display ntp-service ipv6 sessions
Notes: 1 source(master), 2 source(peer), 3 selected, 4 candidate, 5 configured.

 Source:  [1234]3000::2
 Reference: 127.127.1.0          Clock stratum: 2
 Reachabilities: 111             Poll interval: 64
 Last receive time: 23           Offset: -0.0
 Roundtrip delay: 0.0            Dispersion: 0.0

 Total sessions: 1
```

5. Configure Switch B:

   Because Switch A and Switch C are on different subnets, you must enable the IPv6 multicast functions on Switch B before Switch A can receive IPv6 multicast messages from Switch C.

   # Enable IPv6 multicast functions.

   ```
   <SwitchB> system-view
   [SwitchB] ipv6 multicast routing
   [SwitchB-mrib6] quit
   [SwitchB] interface vlan-interface 2
   [SwitchB-Vlan-interface2] ipv6 pim dm
   [SwitchB-Vlan-interface2] quit
   [SwitchB] vlan 3
   [SwitchB-vlan3] port ten-gigabitethernet 1/0/1
   [SwitchB-vlan3] quit
   [SwitchB] interface vlan-interface 3
   [SwitchB-Vlan-interface3] mld enable
   [SwitchB-Vlan-interface3] mld static-group ff24::1
   [SwitchB-Vlan-interface3] quit
   [SwitchB] mld-snooping
   [SwitchB-mld-snooping] quit
   [SwitchB] interface ten-gigabitethernet 1/0/1
   [SwitchB-Ten-GigabitEthernet1/0/1] mld-snooping static-group ff24::1 vlan 3
   ```

6. Configure Switch A:

   # Enable the NTP service.

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

```
[SwitchA] ntp-service enable
```

# Configure Switch A to operate in IPv6 multicast client mode and receive IPv6 multicast messages on VLAN-interface 3.

```
[SwitchA] interface vlan-interface 3
[SwitchA-Vlan-interface3] ntp-service ipv6 multicast-client ff24::1
```

7. Verify the configuration:

   # Verify that Switch A has synchronized to Switch C, and the clock stratum level is 3 on Switch A and 2 on Switch C.

```
[SwitchA-Vlan-interface3] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3000::2
 Local mode: bclient
 Reference clock ID: 165.84.121.65
 Leap indicator: 00
 Clock jitter: 0.165741 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00534 ms
 Root dispersion: 4.51282 ms
 Reference time: d0c61289.10b1193f  Wed, Dec 29 2010 20:03:21.065
```

   # Verify that an IPv6 NTP association has been established between Switch A and Switch C.

```
[SwitchA-Vlan-interface3] display ntp-service ipv6 sessions
Notes: 1 source(master), 2 source(peer), 3 selected, 4 candidate, 5 configured.

 Source:   [124]3000::2
 Reference: 127.127.1.0          Clock stratum: 2
 Reachabilities: 2               Poll interval: 64
 Last receive time: 71           Offset: -0.0
 Roundtrip delay: 0.0            Dispersion: 0.0

 Total sessions: 1
```

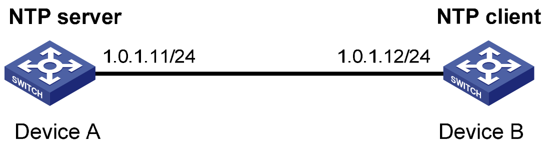
Configuration example for NTP client/server mode with authentication

Network requirements

As shown in Figure 15:

- Configure the local clock of Device A as a reference source, with the stratum level 2.
- Configure Device B to operate in client mode and specify Device A as the NTP server of Device B, with Device B as the client.
- Configure NTP authentication on both Device A and Device B.

Figure 15 Network diagram



Configuration procedure

1. Set the IP address for each interface, and make sure Device A and Device B can reach each other, as shown in Figure 15. (Details not shown.)
2. Configure Device A:

   # Enable the NTP service.

   ```
   <DeviceA> system-view
   [DeviceA] ntp-service enable
   ```

   # Specify the local clock as the reference source, with the stratum level 2.

   ```
   [DeviceA] ntp-service refclock-master 2
   ```
3. Configure Device B:

   # Enable the NTP service.

   ```
   <DeviceB> system-view
   [DeviceB] ntp-service enable
   ```

   # Enable NTP authentication on Device B.

   ```
   [DeviceB] ntp-service authentication enable
   ```

   # Set an authentication key, and input the key in plain text.

   ```
   [DeviceB] ntp-service authentication-keyid 42 authentication-mode md5 simple
   aNiceKey
   ```

   # Specify the key as a trusted key.

   ```
   [DeviceB] ntp-service reliable authentication-keyid 42
   ```

   # Specify Device A as the NTP server of Device B, and associate the server with key 42.

   ```
   [DeviceB] ntp-service unicast-server 1.0.1.11 authentication-keyid 42
   ```

   Before Device B can synchronize its clock to that of Device A, enable NTP authentication for Device A.
4. Configure NTP authentication on Device A:

   # Enable NTP authentication.

   ```
   [DeviceA] ntp-service authentication enable
   ```

   # Set an authentication key, and input the key in plain text.

   ```
   [DeviceA] ntp-service authentication-keyid 42 authentication-mode md5 simple
   aNiceKey
   ```

   # Specify the key as a trusted key.

   ```
   [DeviceA] ntp-service reliable authentication-keyid 42
   ```
5. Verify the configuration:

   # Verify that Device B has synchronized to Device A, and the clock stratum level is 3 on Device B and 2 on Device A.

   ```
   [DeviceB] display ntp-service status
    Clock status: synchronized
    Clock stratum: 3
   ```

```
 System peer: 1.0.1.11
 Local mode: client
 Reference clock ID: 1.0.1.11
 Leap indicator: 00
 Clock jitter: 0.005096 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00655 ms
 Root dispersion: 1.15869 ms
 Reference time: d0c62687.ab1bba7d  Wed, Dec 29 2010 21:28:39.668
```

# Verify that an IPv4 NTP association has been established between Device B and Device A.

```
[DeviceB] display ntp-service sessions
       source          reference       stra reach poll  now offset  delay disper
********************************************************************************
 [1245]1.0.1.11        127.127.1.0        2     1   64  519   -0.0 0.0065    0.0
Notes: 1 source(master),2 source(peer),3 selected,4 candidate,5 configured.
 Total sessions: 1
```

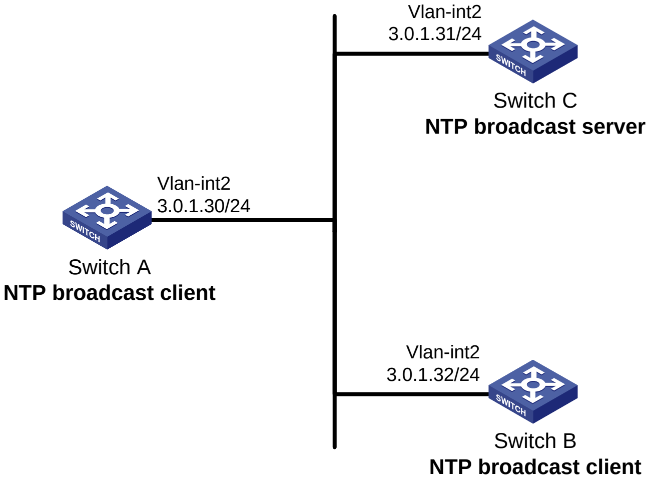
Configuration example for NTP broadcast mode with authentication

Network requirements

As shown in Figure 16, Switch C functions as the NTP server for multiple devices on different network segments and synchronizes the time among multiple devices. Switch A and Switch B authenticate the reference source.

- Configure Switch C's local clock as a reference source, with the stratum level 3.
- Configure Switch C to operate in broadcast server mode and send out broadcast messages from VLAN-interface 2.
- Configure Switch A and Switch B to operate in broadcast client mode and receive broadcast messages through VLAN-interface 2.
- Enable NTP authentication on Switch A, Switch B, and Switch C.

Figure 16 Network diagram



Configuration procedure

1. Set the IP address for each interface, and make sure Switch A, Switch B, and Switch C can reach each other, as shown in Figure 16. (Details not shown.)
2. Configure Switch A:

   # Enable the NTP service.

   ```
   <SwitchA> system-view
   [SwitchA] ntp-service enable
   ```

   # Enable NTP authentication on Switch A. Configure an NTP authentication key, with the key ID of 88 and key value of 123456. Input the key in plain text, and specify it as a trusted key.

   ```
   [SwitchA] ntp-service authentication enable
   [SwitchA] ntp-service authentication-keyid 88 authentication-mode md5 simple 123456
   [SwitchA] ntp-service reliable authentication-keyid 88
   ```

   # Configure Switch A to operate in NTP broadcast client mode and receive NTP broadcast messages on VLAN-interface 2.

   ```
   [SwitchA] interface vlan-interface 2
   [SwitchA-Vlan-interface2] ntp-service broadcast-client
   ```
3. Configure Switch B:

   # Enable the NTP service.

   ```
   <SwitchB> system-view
   [SwitchB] ntp-service enable
   ```

   # Enable NTP authentication on Switch B. Configure an NTP authentication key, with the key ID of 88 and key value of 123456. Input the key in plain text and specify it as a trusted key.

   ```
   [SwitchB] ntp-service authentication enable
   [SwitchB] ntp-service authentication-keyid 88 authentication-mode md5 simple 123456
   [SwitchB] ntp-service reliable authentication-keyid 88
   ```

   # Configure Switch B to operate in broadcast client mode and receive NTP broadcast messages on VLAN-interface 2.

   ```
   [SwitchB] interface vlan-interface 2
   [SwitchB-Vlan-interface2] ntp-service broadcast-client
   ```
4. Configure Switch C:

# Enable the NTP service.

```
<SwitchC> system-view
[SwitchC] ntp-service enable
```

# Specify the local clock as the reference source, with the stratum level 3.

```
[SwitchC] ntp-service refclock-master 3
```

# Configure Switch C to operate in NTP broadcast server mode and use VLAN-interface 2 to send NTP broadcast packets.

```
[SwitchC] interface vlan-interface 2
[SwitchC-Vlan-interface2] ntp-service broadcast-server
[SwitchC-Vlan-interface2] quit
```

5. Verify the configuration:

# NTP authentication is enabled on Switch A and Switch B, but not on Switch C, so Switch A and Switch B cannot synchronize their local clocks to Switch C. Display the NTP service status on Switch B.

```
[SwitchB-Vlan-interface2] display ntp-service status
 Clock status: unsynchronized
 Clock stratum: 16
 Reference clock ID: none
```

6. Enable NTP authentication on Switch C:

# Enable NTP authentication on Switch C. Configure an NTP authentication key, with the key ID of 88 and key value of 123456. Input the key in plain text, and specify it as a trusted key.

```
[SwitchC] ntp-service authentication enable
[SwitchC] ntp-service authentication-keyid 88 authentication-mode md5 simple 123456
[SwitchC] ntp-service reliable authentication-keyid 88
```

# Specify Switch C as an NTP broadcast server, and associate the key 88 with Switch C.

```
[SwitchC] interface vlan-interface 2
[SwitchC-Vlan-interface2] ntp-service broadcast-server authentication-keyid 88
```

7. Verify the configuration:

After NTP authentication is enabled on Switch C, Switch A and Switch B can synchronize their local clocks to Switch C.

# Verify that Switch B has synchronized to Switch C, and the clock stratum level is 4 on Switch B and 3 on Switch C.

```
[SwitchB-Vlan-interface2] display ntp-service status
 Clock status: synchronized
 Clock stratum: 4
 System peer: 3.0.1.31
 Local mode: bclient
 Reference clock ID: 3.0.1.31
 Leap indicator: 00
 Clock jitter: 0.006683 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00127 ms
 Root dispersion: 2.89877 ms
 Reference time: d0d287a7.3119666f  Sat, Jan  8 2011  6:50:15.191
```

# Verify that an IPv4 NTP association has been established between Switch B and Switch C.

```
[SwitchB-Vlan-interface2] display ntp-service sessions
       source           reference       stra reach poll  now offset  delay disper
********************************************************************************
 [1245]3.0.1.31        127.127.1.0        3     3   64   68   -0.0 0.0000    0.0
Notes: 1 source(master),2 source(peer),3 selected,4 candidate,5 configured.
 Total sessions: 1
```

Configuring SNTP

SNTP is a simplified, client-only version of NTP specified in RFC 4330. SNTP supports only the client/server mode. An SNTP-enabled device can receive time from NTP servers, but cannot provide time services to other devices.

SNTP uses the same packet format and packet exchange procedure as NTP, but provides faster synchronization at the price of time accuracy.

If you specify multiple NTP servers for an SNTP client, the server with the best stratum is selected. If multiple servers are at the same stratum, the NTP server whose time packet is first received is selected.

Configuration restrictions and guidelines

Follow these restrictions and guidelines when you configure SNTP:

- You cannot configure both NTP and SNTP on the same device.
- Make sure you use the **clock protocol** command to specify the time protocol as NTP.

Configuration task list

Tasks at a glance
(Required.) Enabling the SNTP service
(Required.) Specifying an NTP server for the device
(Optional.) Configuring SNTP authentication

Enabling the SNTP service

The NTP service and SNTP service are mutually exclusive. You can only enable either NTP service or SNTP service at a time.

To enable the SNTP service:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable the SNTP service.	**sntp enable**	By default, the SNTP service is not enabled.

Specifying an NTP server for the device

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Specify an NTP server for the device.	- For IPv4: **sntp unicast-server** { *server-name* | *ip-address* } [**authentication-keyid** *keyid* | **source** *interface-type interface-number* | **version** *number*] * - For IPv6: **sntp ipv6 unicast-server** { *server-name* | *ipv6-address* } [**authentication-keyid** *keyid* | **source** *interface-type interface-number*] *	By default, no NTP server is specified for the device. Repeat this step to specify multiple NTP servers. To use authentication, you must specify the **authentication-keyid** *keyid* option.

To use an NTP server as the time source, make sure its clock has been synchronized. If the stratum level of the NTP server is greater than or equal to that of the client, the client does not synchronize with the NTP server.

Configuring SNTP authentication

SNTP authentication makes sure an SNTP client is synchronized only to an authenticated trustworthy NTP server.

To make sure SNTP authentication can work, follow these guidelines on configuring SNTP authentication:

- Enable authentication on both the NTP server and the SNTP client.
- Configure the SNTP client with the same authentication key ID and key value as the NTP server, and specify the key as a trusted key on both the NTP server and the SNTP client. For information about configuring NTP authentication on an NTP server, see "Configuring NTP."
- Associate the specified key with an NTP server on the SNTP client.

With authentication disabled, the SNTP client can synchronize with the NTP server regardless of whether the NTP server is enabled with authentication.

To configure SNTP authentication on the SNTP client:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable SNTP authentication.	**sntp authentication enable**	By default, SNTP authentication is disabled.
3. Configure an SNTP authentication key.	**sntp authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	By default, no SNTP authentication key is configured.

Step	Command	Remarks
4. Specify the key as a trusted key.	**sntp reliable authentication-keyid** *keyid*	By default, no trusted key is specified.
5. Associate the SNTP authentication key with an NTP server.	• For IPv4: **sntp unicast-server** { *server-name* | *ip-address* } **authentication-keyid** *keyid* • For IPv6: **sntp ipv6 unicast-server** { *server-name* | *ipv6-address* } **authentication-keyid** *keyid*	By default, no NTP server is specified.

Displaying and maintaining SNTP

Execute **display** commands in any view.

Task	Command
Display information about all IPv6 SNTP associations.	**display sntp ipv6 sessions**
Display information about all IPv4 SNTP associations.	**display sntp sessions**

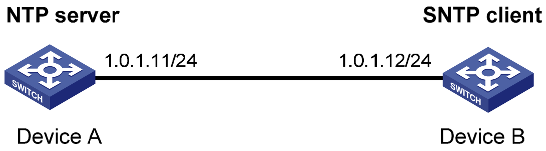
SNTP configuration example

Network requirements

As shown in Figure 17:

- Configure the local clock of Device A as a reference source, with the stratum level 2.
- Configure Device B to operate in SNTP client mode, and specify Device A as the NTP server.
- Configure NTP authentication on Device A and SNTP authentication on Device B.

Figure 17 Network diagram



Configuration procedure

1. Set the IP address for each interface, and make sure Device A and Device B can reach each other, as shown in Figure 17. (Details not shown.)
2. Configure Device A:

   # Enable the NTP service.

   ```
   <DeviceA> system-view
   [DeviceA] ntp-service enable
   ```

   # Configure the local clock of Device A as a reference source, with the stratum level 2.

```
[DeviceA] ntp-service refclock-master 2
```

# Enable NTP authentication on Device A.

```
[DeviceA] ntp-service authentication enable
```

# Configure an NTP authentication key, with the key ID of **10** and key value of **aNiceKey**. Input the key in plain text.

```
[DeviceA] ntp-service authentication-keyid 10 authentication-mode md5 simple
aNiceKey
```

# Specify the key as a trusted key.

```
[DeviceA] ntp-service reliable authentication-keyid 10
```

3. Configure Device B:

# Enable the SNTP service.

```
<DeviceB> system-view
[DeviceB] sntp enable
```

# Enable SNTP authentication on Device B.

```
[DeviceB] sntp authentication enable
```

# Configure an SNTP authentication key, with the key ID of **10** and key value of **aNiceKey**. Input the key in plain text.

```
[DeviceB] sntp authentication-keyid 10 authentication-mode md5 simple aNiceKey
```

# Specify the key as a trusted key.

```
[DeviceB] sntp reliable authentication-keyid 10
```

# Specify Device A as the NTP server of Device B, and associate the server with key 10.

```
[DeviceB] sntp unicast-server 1.0.1.11 authentication-keyid 10
```

4. Verify the configuration:

# Verify that an SNTP association has been established between Device B and Device A, and Device B has synchronized to Device A.

```
[DeviceB] display sntp sessions
NTP server     Stratum   Version    Last receive time
1.0.1.11        2         4          Tue, May 17 2011  9:11:20.833 (Synced)
```

Configuring PTP

Overview

Precision Time Protocol (PTP) synchronizes time among devices. It provides greater accuracy than other time synchronization protocols such as NTP. For more information about NTP, see "Configuring NTP"

Basic concepts

PTP profile

A PTP profile defines two PTP standards: IEEE 1588 version 2 and IEEE 802.1AS.

- **IEEE 1588 version 2**—1588v2 defines high-accuracy clock synchronization mechanisms. It can be customized, enhanced, or tailored as needed. 1588v2 is the latest version.
- **IEEE 802.1AS**—802.1AS is introduced based on IEEE 1588. It specifies a profile for use of IEEE 1588-2008 for time synchronization over a virtual bridged local area network (as defined by IEEE 802.1Q). The BMC algorithm in 802.1AS supports only the peer delay mode, and point-to-point full-duplex Ethernet, IEEE 802.11, and IEEE 802.3 EPON links.

PTP domain

A PTP domain refers to a network that is enabled with PTP. A PTP domain has only one reference clock called "grandmaster clock (GM)." All devices in the domain synchronize to the clock.

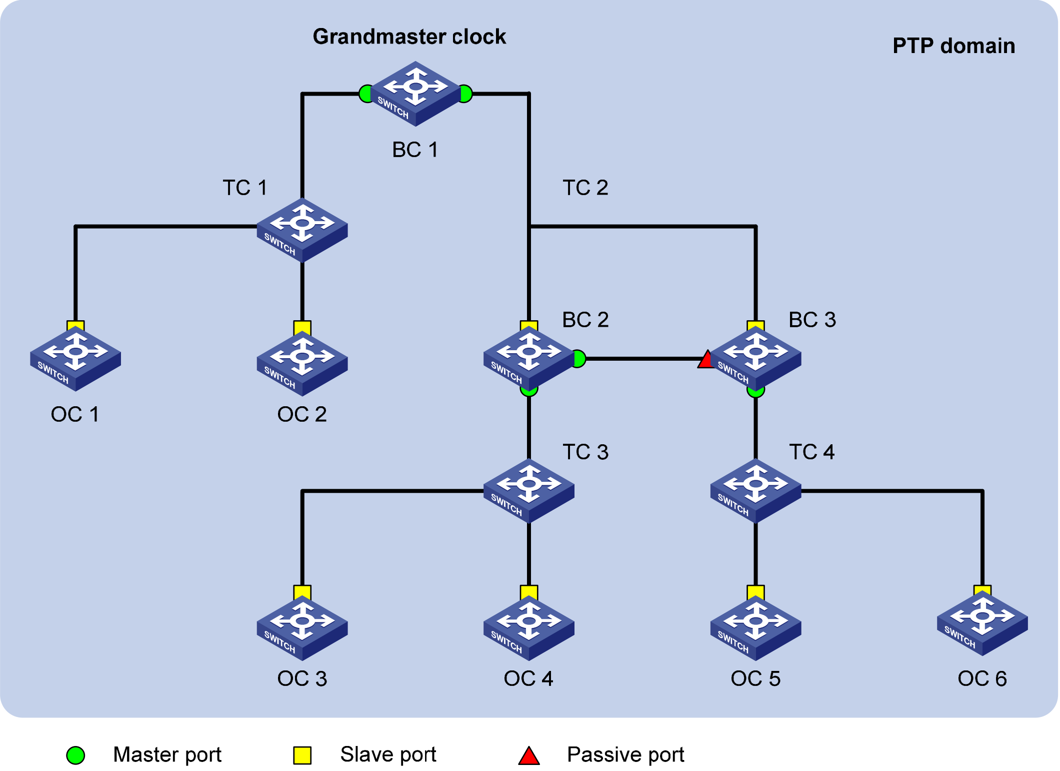
Clock node and PTP port

A node in a PTP domain is a clock node. A port enabled with PTP is a PTP port. PTP defines the following three types of basic clock nodes:

- **Ordinary Clock (OC)**—A PTP clock with a single PTP port in a PTP domain for time synchronization. It synchronizes time from its upstream clock node through the port. If a clock node works as the clock source and sends synchronization time through a single PTP port to its downstream clock node, it is also called an OC.
- **Boundary Clock (BC)**—A clock with more than one PTP port in a PTP domain for time synchronization. A BC uses one of the ports to synchronize time from its upstream clock node, and uses the other ports to synchronize time to the relevant upstream clock nodes. If a clock node works as the clock source and synchronizes time through multiple PTP ports to its downstream clock nodes, it is also called a BC, such as BC 1 in Figure 18.
- **Transparent Clock (TC)**—A TC does not need to keep time consistency with other clock nodes. A TC has multiple PTP ports. It only forwards PTP messages among these ports and performs delay corrections for the messages, instead of performing time synchronization. TCs include the following types:
  - **End-to-End Transparent Clock (E2ETC)**—Forwards non-P2P packets in the network and calculates the delay of the entire link.
  - **Peer-to-Peer Transparent Clock (P2PTC)**—Forwards only Sync, Follow_Up, and Announce messages, terminates other PTP messages, and calculates the delay of each link segment.

Figure 18 shows the positions of these three types of clock nodes in a PTP domain.

Figure 18 Clock nodes in a PTP domain



Besides the three basic types of clock nodes, PTP introduces some hybrid clock nodes. For example, a TC+OC has multiple PTP ports in a PTP domain: one port is the OC type, and the others are the TC type. A TC+OC forwards PTP messages through TC-type ports and performs delay corrections. In addition, it synchronizes time through its OC-type port. TC+OCs include these types: E2ETC+OC and P2PTC+OC.

Master-member/subordinate relationship

The master-member/subordinate relationship is defined as follows:

- **Master/Member node**—A master node sends a synchronization message, and a member node receives the synchronization message.
- **Master/Member clock**—The clock on a master node is a master clock, and that on a member node is a member clock.
- **Master/Subordinate port**—A master port sends a synchronization message, and a subordinate port receives the synchronization message. The master and subordinate ports can be on a BC or an OC. A port that neither receives nor sends synchronization time is a passive port.

Grandmaster clock

In Figure 18, all clock nodes are organized together and ultimately derive their time from a clock known as the "grandmaster clock (GM)." The GM clock source synchronizes its time to the entire PTP domain through PTP messages exchanged among the clock nodes.

A GM can be manually configured, or it can be elected through the Best Master Clock (BMC) algorithm as follows:

1. By exchanging announce messages containing the priorities, time class, and time accuracy of GMs, clock nodes in a PTP domain elect a GM. The master nodes, member nodes, master ports, and subordinate ports are specified during the process. Then a loop-free, interconnected spanning tree with the GM as the root is generated for the PTP domain.

2. The master node periodically sends announce messages to member nodes. If the member nodes do not receive announce messages from the master node, they consider the master node invalid and start to elect another GM.

Clock source type

A clock node of a device can use one of the following clock sources: local clock source and BITS clock source (BITS1 and BITS2) that connects the device.

- **Local clock**—38.88 MHz clock signals generated by a crystal oscillator inside the clock monitoring module.
- **BITS clock**—Clock signals generated by a BITS clock device. The signals are sent to the clock monitoring module through a specific interface on the device and then sent to all member devices by the clock monitoring module.

The clock node determines to use which type of clock source based on the specified algorithm.

Synchronization mechanism

Based on the exchanged synchronization messages, a member node calculates the round-trip delay of the path to the master node. The one-way delay equals half of the round-trip delay (assume the delays in both directions are the same). Then the member node synchronizes its clock with the master clock according to the offset between the clocks.

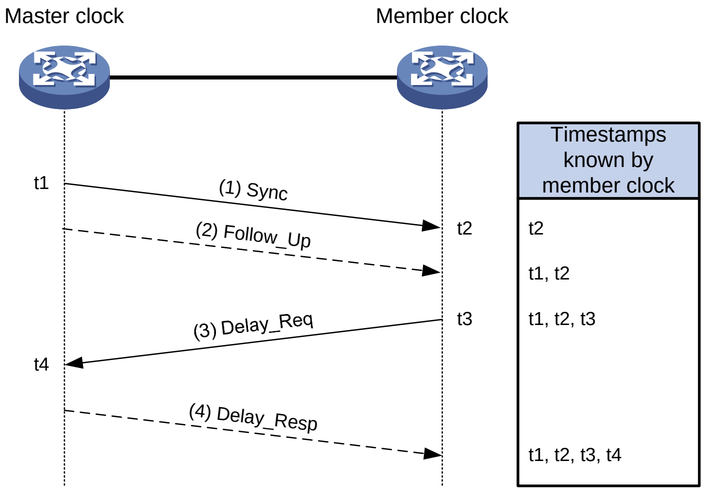
PTP defines the following transmission delay measurement mechanisms:

- Request_Response.
- Peer Delay.

The basis of the two mechanisms is that the transmission delay from the master clock to the member clock is the same as that from the member clock to the master clock.

Request_Response

Figure 19 Operation procedure of the Request_Response mechanism



Timestamps known by member clock
t2
t1, t2
t1, t2, t3
t1, t2, t3, t4

Figure 19 shows the operation procedure of the Request_Response mechanism, which applies only to peer-to-peer delay measurement. It shows an example of the Request_Response mechanism in two-step mode.

1. The master clock sends a Sync message to the member clock, and records the sending time t1. Upon receiving the message, the member clock records the receiving time t2.
2. After sending the Sync message, the master clock immediately sends a Follow_Up message carrying time t1.
3. The member clock sends a Delay_Req message to calculate the transmission delay in the reverse direction, and records the sending time t3. Upon receiving the message, the master clock records the receiving time t4.
4. The master clock returns a Delay_Resp message carrying time t4.

From the above process, the member clock collects four timestamps, t1 to t4, and obtains the round-trip delay to the master clock by using the following calculation:

- [(t2 – t1) + (t4 – t3)]

The member clock also obtains the one-way delay by using the following calculation:

- [(t2 – t1) + (t4 – t3)] / 2

The offset between the member and master clocks is obtained by using the following calculation:

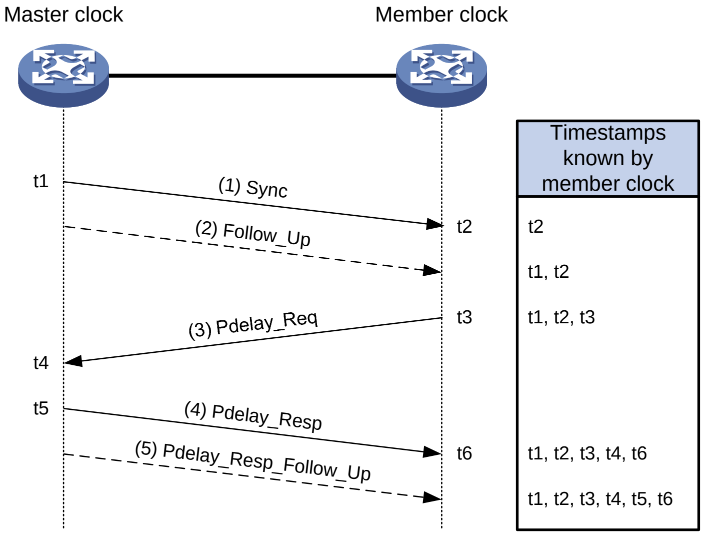
- (t2 – t1) – [(t2 – t1) + (t4 – t3)] / 2
- [(t2 – t1) – (t4 – t3)] / 2

Depending on whether to send Follow_Up messages, the Request_Response mechanism includes two modes: single-step and two-step.

- In single-step mode, t1 is carried in the Sync message, and no Follow_Up message is sent.
- In two-step mode, t1 is carried in the Follow_Up message.

Peer Delay

Figure 20 Operation procedure of the Peer Delay mechanism



The Peer Delay mechanism uses Pdelay messages to calculate link delay, which applies only to point-to-point delay measurement. Figure 20 shows an example of the Peer Delay mechanism by using the two-step mode.

1. The master clock sends a Sync message to the member clock, and records the sending time t1. Upon receiving the message, the member clock records the receiving time t2.

2. After sending the Sync message, the master clock sends a Follow_Up message carrying time t1 immediately.
3. The member clock sends a Pdelay_Req message to calculate the transmission delay in the reverse direction, and records the sending time t3. Upon receiving the message, the master clock records the receiving time t4.
4. The master clock returns a Pdelay_Resp message carrying time t4, and records the sending time t5. Upon receiving the message, the member clock records the receiving time t6.
5. After sending the Pdelay_Resp message, the master clock sends a Pdelay_Resp_Follow_Up message carrying time t5 immediately.

From the above process, the member clock collects six timestamps, t1 to t6, and obtains the round-trip delay to the master clock by using the following calculation:

- [(t4 – t3) + (t6 – t5)]

  The member clock also obtains the one-way delay by using the following calculation:

- [(t4 – t3) + (t6 – t5)] / 2

  The offset between the member and master clocks is as follows:

- (t2 – t1) – [(t4 – t3) + (t6 – t5)] / 2

Depending on whether to send Follow_Up messages, the Peer Delay mechanism includes two modes: single-step and two-step.

- In single-step mode, t1 is carried in the Sync message, and no Follow_Up message is sent. The offset between t5 and t4 is carried in the Pdelay_Resp message, and no Pdelay_Resp_Follow_Up message is sent.
- In two-step mode, t1 is carried in the Follow_Up message, and t4 and t5 are carried in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages.

Protocols and standards

- IEEE 1588-2008, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*
- IEEE P802.1AS, *Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*

Configuring clock nodes

Before performing the following configurations, define the scope of the PTP domain and the role of every clock node.

Configuration task list

Tasks at a glance
(Required.) Specifying a PTP standard

Tasks at a glance
The PTP standard is IEEE 1588 version 2: (Required.) Specifying the clock node type (Optional.) Specifying a PTP domain (Optional.) Configuring an OC to operate as only a member clock (Optional.) Configuring the role of a PTP port (Optional.) Configuring the mode for carrying timestamps (Optional.) Specifying a delay measurement mechanism for a BC or OC (Optional.) Configuring the port type for a TC+OC (Optional.) Configuring the interval for sending announce messages (Optional.) Configuring the interval for sending Pdelay_Req messages (Optional.) Configuring the interval for sending Sync messages (Optional.) Configuring the minimum interval for sending Delay_Req messages (Optional.) Configuring the MAC address for non-pdelay messages (Optional.) Specifying the protocol for encapsulating PTP messages as UDP (IPv4) (Optional.) Specifying the source IP address for UDP packets (Optional.) Configuring the delay correction value (Optional.) Configuring the cumulative offset between the UTC and TAI (Optional.) Configuring the correction date of the UTC (Optional.) Configuring the parameters of the BITS clock (Optional.) Configuring a priority of the clock (Optional.) Specifying the system time source as PTP (Required.) Enabling PTP on a port
The PTP standard is IEEE 802.1AS (802.1AS): (Required.) Specifying the clock node type (Optional.) Specifying a PTP domain (Optional.) Configuring an OC to operate as only a member clock (Optional.) Configuring the role of a PTP port (Optional.) Configuring the port type for a TC+OC (Optional.) Configuring the interval for sending announce messages (Optional.) Configuring the interval for sending Pdelay_Req messages (Optional.) Configuring the interval for sending Sync messages (Optional.) Configuring the minimum interval for sending Delay_Req messages (Optional.) Configuring the delay correction value (Optional.) Configuring the cumulative offset between the UTC and TAI (Optional.) Configuring the correction date of the UTC (Optional.) Configuring the parameters of the BITS clock (Optional.) Configuring a priority of the clock (Optional.) Specifying the system time source as PTP (Required.) Enabling PTP on a port

Specifying a PTP standard

Before configuring PTP, specify a PTP standard first. Otherwise, PTP cannot operate. Changing the PTP standard for the device clears all PTP configurations defined by the standard.

To specify a PTP standard:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Specify a PTP standard.	**ptp profile** { **1588v2** | **8021as** }	By default, no PTP standard is configured, and PTP is not running on the device.

Specifying the clock node type

You can configure only one of the following six types of clock nodes for a device: OC, BC, E2ETC, P2PTC, E2ETC+OC, or P2PTC+OC.

Follow these guidelines when you specify the clock node type:

- Before specifying the clock node type, specify a PTP standard first.
- If the PTP standard is IEEE 802.1AS, the clock node type cannot be E2ETC or E2ETC+OC.
- Changing the clock node type clears all PTP configurations except the PTP standard.

To specify the clock node type:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Specify the clock node type for the device.	**ptp mode** { **bc** | **e2etc** | **e2etc-oc** | **oc** | **p2ptc** | **p2ptc-oc** }	By default, no clock node type is specified.

Specifying a PTP domain

Within a PTP domain, all devices follow the same rules to communicate with each other. Devices in different PTP domains cannot communicate with each other.

To specify a PTP domain:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Specify a PTP domain for the device.	**ptp domain** *value*	By default, PTP devices are in PTP domain 0.

Configuring an OC to operate as only a member clock

Typically an OC can work either as a master clock to send synchronization messages or a member clock to receive synchronization messages. This task allows you to configure an OC to operate as only a member clock.

This task is applicable only to OCs.

This configuration is automatically cleared after you change the clock node type for the device.

If an OC is operating as only a member clock, you can also use the **ptp force-state** command to configure its PTP port as a master port or passive port.

To configure an OC to operate as only a member clock:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Configure the OC to operate as only a member clock.	**ptp slave-only**	By default, the OC is not configured to operate as only a member clock.

Configuring the role of a PTP port

Typically the master/member relationships are automatically specified through BMC. This task allows you to manually configure the master/member relationships among clock nodes. The **ptp force-state** command is available only after you configure the **ptp active force-state** command.

Follow these guidelines when you configure the role of a PTP port:

- Only one subordinate port is allowed to be configured for a device.
- This task is also applicable to an OC that operates in **slave-only** mode.

To configure the PTP port role on an OC, BC, E2ETC+OC, or P2PTC+OC:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view	**interface** *interface-type interface-number*	N/A
3.	Configure the role of the PTP port.	**ptp force-state** { **master** | **passive** | **slave** }	By default, the PTP port role is automatically specified through BMC.
4.	Quit interface view.	**quit**	N/A
5.	Activate the port role configuration.	**ptp active force-state**	By default, the port role configuration is not activated.

Configuring the mode for carrying timestamps

Timestamps can be carried in either of the following two modes:

- **Single-step mode**—In single-step mode, the Sync message in the Request_Response and Peer Delay mechanisms and the Pdelay_Resp message in the Peer Delay mechanism carry the sending time of the messages.
- **Two-step mode**—In two-step mode, the Sync message in the Request_Response and Peer Delay mechanisms and the Pdelay_Resp message in the Peer Delay mechanism do not carry the sending time of the messages. The sending time is carried in other messages.

To configure the mode for carrying timestamps for every clock node:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3. Configure the mode for carrying timestamps.	**ptp clock-step** { **one-step** | **two-step** }	By default, two-step mode is adopted.

Specifying a delay measurement mechanism for a BC or OC

PTP defines two transmission delay measurement mechanisms: Request_Response and Peer Delay. Ports on the same link must share the same delay measurement mechanism. Otherwise, they cannot communicate with one another. BCs and OCs do not have a default delay measurement mechanism. You must specify a delay measurement mechanism for them. The delay measurement mechanism of E2ETCs and E2ETC+OCs is Request_Response, and that of P2PTCs and P2PTC+OCs is Peer Delay. You cannot change these defaults.

This task is applicable only to BCs and OCs.

If the PTP standard is IEEE 802.1AS, only Peer Delay mode is supported.

To specify a delay measurement mechanism for a BC or OC:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3. Specify a delay measurement mechanism for a BC or OC.	**ptp delay-mechanism** { **e2e** | **p2p** }	By default, the delay measurement mechanism depends on the PTP standard.

Configuring the port type for a TC+OC

All ports on a TC+OC (E2ETC+OC or P2PTC+OC) are TCs by default. This command allows you to configure one of the ports as an OC. This task is applicable only to E2ETC+OCs and P2PTC+OCs.

When a TC+OC is synchronizing time to a downstream clock node through a TC, do not enable an OC to synchronize time from the downstream clock node. Otherwise, time synchronization might be affected.

To configure the port type for a TC+OC as OC:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the port type for a TC+OC as OC.	**ptp port-mode oc**	By default, the type of all ports on a TC+OC is TC.

Configuring the interval for sending announce messages

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the interval for sending announce messages.	**ptp announce-interval** *value*	By default: • The interval is 2 (2^1) seconds if the PTP standard is IEEE 1588 version 2. • The interval is 1 (2^0) second if the PTP standard is IEEE 802.1AS.

Specifying the number of announcement intervals before the receiving node stops receiving announce messages

A master node periodically sends announce messages to the member nodes. If a member node does not receive any announce message from the master node within the specified interval, it considers the master node invalid.

If the PTP standard is IEEE 1588 version 2, the interval is the announce message sending interval × *multiple-value*. If the PTP standard is IEEE 802.1AS, the interval is the announce message sending interval for the master node × *multiple-value*.

To specify the number of announcement intervals before the receiving node stops receiving announce messages:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Specify the number of announcement intervals before the receiving node stops receiving announce messages.	**ptp announce-timeout** *multiple-value*	The default is 3.

Configuring the interval for sending Pdelay_Req messages

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the interval for sending Pdelay_Req messages.	**ptp pdelay-req-interval** *value*	Optional. The default is 1 (2^0) second.

Configuring the interval for sending Sync messages

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the interval for sending Sync messages.	**ptp syn-interval** *value*	By default: • The interval is 1 (2^0) second if the PTP standard is IEEE 1588 version 2. • The interval is 1/8 (2^{-3}) seconds if the PTP standard is IEEE 802.1AS.

Configuring the minimum interval for sending Delay_Req messages

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the minimum interval for sending Delay_Req messages.	**ptp min-delayreq-interval** *value*	The default is 1 (2^0) second. When receiving a Sync or Follow_Up message, an interface can send Delay_Req messages only when the minimum interval is reached.

Configuring the MAC address for non-pdelay messages

Pdelay messages include Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages. The destination MAC address of Pdelay messages is 0180-C200-000E by default, which cannot be modified. The destination MAC address of non-Pdelay messages is either 0180-C200-000E or 011B-1900-0000.

If ports on the same link forward PTP packets of the same type to different destination MAC addresses, they do not receive the packets from each other. You need to configure the same destination MAC address for the ports.

To configure the destination MAC address for non-Pdelay messages on every clock node:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the destination MAC address for non-Pdelay messages.	**ptp destination-mac** *mac-address*	The default is 011B-1900-0000.

Specifying the protocol for encapsulating PTP messages as UDP (IPv4)

PTP messages can be encapsulated in IEEE 802.3/Ethernet packets or UDP packets.

To configure the protocol for encapsulating PTP messages as UDP (IPv4):

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the protocol for encapsulating PTP messages as UDP (IPv4).	**ptp transport-protocol udp**	By default, PTP messages are encapsulated in IEEE 802.3/Ethernet packets.

Specifying the source IP address for UDP packets

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter interface view.	**ptp source** *ip-address*	By default, no source IP address is specified for PTP messages encapsulated in UDP packets.

Configuring the delay correction value

PTP performs time synchronization based on the assumption that the delays in sending and receiving messages are the same. However, this is not practical. If you know the offset between the delays in sending and receiving messages, you can configure the delay correction value for more accurate time synchronization.

To configure the delay correction value for every clock node:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure delay correction value.	**ptp asym-correction** { **minus** | **plus** } *value*	Optional. The default is 0 nanoseconds, which means delay correction is not performed.

Configuring the cumulative offset between the UTC and TAI

The time displayed on a device is based on the Coordinated Universal Time (UTC). There is an offset between UTC and TAI (International Atomic Time in English), which is made public periodically. This task allows you to adjust the offset between the UTC and TAI on the device. It is applicable only to the GM.

To configure the cumulative offset between the UTC and TAI:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Configure the cumulative offset between the UTC and TAI.	**ptp utc offset** *utc-offset*	The default is 0 seconds.

Configuring the correction date of the UTC

This task allows you to adjust the UTC at the last minute (23:59) of the specified date.

To configure the correction date of the UTC:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Configure the correction date of the UTC.	**ptp utc** { **leap59-date** | **leap61-date** } *date*	By default, the correction date of the UTC is not configured. This command takes effect only on the GM.

Configuring the parameters of the BITS clock

Clock nodes in a PTP domain exchange announce messages through BMC to elect a GM. They compare the parameters in the announce messages in the following sequence: priority 1, time class, time accuracy, and priority 2. If all these parameters are the same, the clock node with a smaller port ID (consisting of clock number and port number) wins.

To configure the clock parameters

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure priority 1 of the clock.	**ptp priority clock-source** { **bits1** | **bits2** | **local** } **priority1** *pri1-value*	Optional. The default is 128.
3. Configure the parameters of the BITS clock.	**ptp clock-source** { **bits1** | **bits2** } { **accuracy** *acc-value* | **class** *class-value* | **time-source** *ts-value* }	By default, the time accuracy is 254, the time class is 248, and the attribute value is 160 for the BITS clock.
4. Configure the time accuracy of the BITS clock.	**ptp clock-source** { **bits1** | **bits2** } **accuracy** *acc-value*	Optional. The default is 254.
5. Configure priority 2 of the clock.	**ptp priority clock-source** { **bits1** | **bits2** | **local** } **priority2** *pri2-value*	Optional. The default is 128.
6. Configure the attribute value of the BITS clock.	**ptp clock-source** { **bits1** | **bits2** } **time-source** *ts-value*	Optional. The default is 160.

Configuring a priority of the clock

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure priority 1 of the clock.	**ptp priority clock-source** { **bits1** | **bits2** | **local** } { **priority1** *pri1-value* | **priority2** *pri2-value* }	By default: • If the PTP profile is IEEE 1588 version 2, the default value for both priority 1 and priority 2 is 128. • If the PTP profile is IEEE 802.1AS, the default value is 246 for priority 1 and 248 for priority 2.

Specifying the system time source as PTP

Before the configuration, make sure you use the **clock protocol** command to specify the time protocol as PTP. For more information about the **clock protocol** command, see *Fundamentals Command Reference*.

To specify the system time source as PTP:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Specify the system time source as PTP.	**clock protocol ptp**	By default, the system time source is NTP.

Enabling PTP on a port

A port enabled with PTP becomes a PTP port.

An OC can have only one PTP port.

To enable PTP on a port:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3. Enabling PTP on the port.	**ptp enable**	By default, PPP is disabled on a port.

Displaying and maintaining PTP

Execute **display** commands in any view and the **reset** command in user view.

Task	Command
Display PTP clock information.	**display ptp clock**
Display the delay correction history.	**display ptp corrections**
Display information about foreign master nodes.	**display ptp foreign-masters-record** [**interface** *interface-type interface-number*]
Display PTP information on an interface.	**display ptp interface** [*interface-type interface-number* | **brief**]
Display PTP statistics.	**display ptp** [**brief** | **interface** *interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]
Display PTP clock time properties.	**display ptp time-property**
Clear PTP statistics.	**reset ptp statistics** [**interface** *interface-type interface-number*]

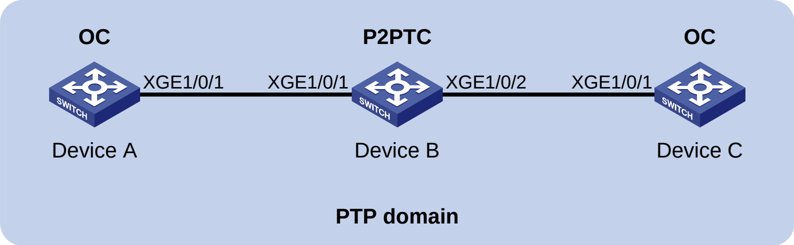
PTP configuration example (IEEE 1588 version 2)

Network requirements

As shown in Figure 21, a PTP domain comprises Device A, Device B, and Device C.

- Configure all devices to use PTP standard IEEE 1588 version 2.
- Configure the clock node type of Device A and Device C as OC, and that of Device B as P2PTC. All clock nodes elect a GM through BMC based on their respective default GM attributes.
- Configure the delay measurement mechanism for Device A and Device C as **p2p**.

Figure 21 Network diagram



Configuration procedure

1. Configure Device A:

   # Specify the PTP standard as IEEE 1588 version 2.

```
<DeviceA> system-view
[DeviceA] ptp profile 1588v2
```

   # Specify the clock node type as OC.

```
[DeviceA] ptp mode oc
```

   # On Ten-GigabitEthernet 1/0/1, specify the delay measurement mechanism as **p2p**, and enable PTP.

```
[DeviceA] interface ten-gigabitethernet 1/0/1
[DeviceA-Ten-GigabitEthernet1/0/1] ptp delay-mechanism p2p
[DeviceA-Ten-GigabitEthernet1/0/1] ptp enable
[DeviceA-Ten-GigabitEthernet1/0/1] quit
```

2. Configure Device B:

   # Specify the PTP standard as IEEE 1588 version 2.

```
<DeviceB> system-view
[DeviceB] ptp profile 1588v2
```

   # Specify the clock node type as P2PTC.

```
[DeviceB] ptp mode p2ptc
```

   # Enable PTP for Ten-GigabitEthernet 1/0/1.

```
[DeviceB] interface ten-gigabitethernet 1/0/1
[DeviceB-Ten-GigabitEthernet1/0/1] ptp enable
[DeviceB-Ten-GigabitEthernet1/0/1] quit
```

   # Enable PTP for Ten-GigabitEthernet 1/0/2.

```
[DeviceB] interface ten-gigabitethernet 1/0/2
[DeviceB-Ten-GigabitEthernet1/0/2] ptp enable
[DeviceB-Ten-GigabitEthernet1/0/2] quit
```

3. Configure Device C:

   # Specify the PTP standard as IEEE 1588 version 2.

```
<DeviceC> system-view
[DeviceC] ptp profile 1588v2
```

# Specify the clock node type as OC.

```
[DeviceC] ptp mode oc
```

# On Ten-GigabitEthernet 1/0/1, specify the delay measurement mechanism as **p2p**, and enable PTP.

```
[DeviceC] interface ten-gigabitethernet 1/0/1
[DeviceC-Ten-GigabitEthernet1/0/1] ptp delay-mechanism p2p
[DeviceC-Ten-GigabitEthernet1/0/1] ptp enable
[DeviceC-Ten-GigabitEthernet1/0/1] quit
```

4. Verify the configuration:

# Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP profile         : IEEE 1588 Version 2
PTP mode            : OC
Slave only          : No
Clock ID            : 000FE2-FFFE-FF0000
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 1
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                  : 248
 Accuracy               : 254
 Offset (log variance) : 65535
Offset from master : 0 (ns)
Mean path delay    : 0 (ns)
Steps removed      : 0
Local clock time   : Sun Jan 15 20:57:29 2011
```

# Display brief PTP statistics on Device A.

```
[DeviceA] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
XGE1/0/1       Master        P2P              Two         0
```

# Display PTP clock information on Device B.

```
[DeviceB] display ptp clock
PTP profile         : IEEE 1588 Version 2
PTP mode            : P2PTC
Slave only          : No
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                  : 248
 Accuracy               : 254
 Offset (log variance) : 65535
Offset from master : N/A
Mean path delay    : N/A
Steps removed      : N/A
Local clock time   : Sun Jan 15 20:57:29 2011
```

# Display brief PTP statistics on Device B.

```
[DeviceB] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
XGE1/0/1      N/A            P2P              Two          0
XGE1/0/2      N/A            P2P              Two          0
```

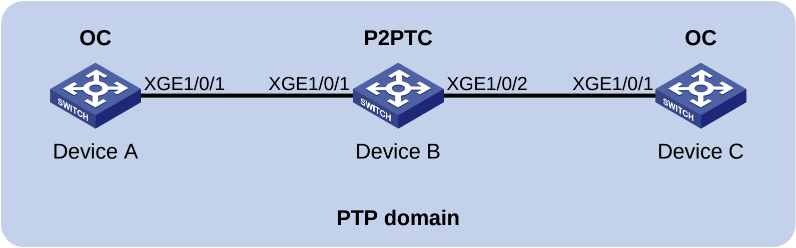
PTP configuration example (IEEE 802.1AS)

Network requirements

As shown in Figure 21, a PTP domain comprises Device A, Device B, and Device C.

- Configure all devices to use PTP standard IEEE 802.1AS.
- Configure the clock node type of Device A and Device C as OC, and that of Device B as P2PTC. All clock nodes elect a GM through BMC based on their respective default GM attributes.
- Configure the delay measurement mechanism for Device A and Device C as **p2p**.

Figure 22 Network diagram



Configuration procedure

1. Configure Device A:

# Specify the PTP standard as IEEE 802.1AS.

```
<DeviceA> system-view
[DeviceA] ptp profile 802.1AS
```

# Specify the clock node type as OC.

```
[DeviceA] ptp mode oc
```

# Enable PTP on Ten-GigabitEthernet 1/0/1.

```
[DeviceA] interface ten-gigabitethernet 1/0/1
[DeviceA-Ten-GigabitEthernet1/0/1] ptp enable
[DeviceA-Ten-GigabitEthernet1/0/1] quit
```

2. Configure Device B:

# Specify the PTP standard as IEEE 802.1AS.

```
<DeviceB> system-view
[DeviceB] ptp profile 802.1AS
```

# Specify the clock node type as P2PTC.

```
[DeviceB] ptp mode p2ptc
```

# Enable PTP for Ten-GigabitEthernet 1/0/1.

```
[DeviceB] interface ten-gigabitethernet 1/0/1
[DeviceB-Ten-GigabitEthernet1/0/1] ptp enable
[DeviceB-Ten-GigabitEthernet1/0/1] quit
```

# Enable PTP for Ten-GigabitEthernet 1/0/2.

```
[DeviceB] interface ten-gigabitethernet 1/0/2
[DeviceB-Ten-GigabitEthernet1/0/2] ptp enable
[DeviceB-Ten-GigabitEthernet1/0/2] quit
```

3. Configure Device C:

# Specify the PTP standard as IEEE 1588 802.1AS.

```
<DeviceC> system-view
[DeviceC] ptp profile 802.1AS
```

# Specify the clock node type as OC.

```
[DeviceC] ptp mode oc
```

# Enable PTP on Ten-GigabitEthernet 1/0/1.

```
[DeviceC] interface ten-gigabitethernet 1/0/1
[DeviceC-Ten-GigabitEthernet1/0/1] ptp enable
[DeviceC-Ten-GigabitEthernet1/0/1] quit
```

4. Verify the configuration:

When the network is stable, perform the following tasks:

- Use the **display ptp clock** command to display PTP clock information.
- Use the **display ptp interface brief** command to display brief PTP statistics on an interface.

# Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP profile          : IEEE 802.1AS
PTP mode             : OC
Slave only           : No
Clock ID             : 000FE2-FFFE-FF0000
Clock type           : Local
Clock domain         : 0
Number of PTP ports : 1
Priority1      : 246
Priority2      : 248
Clock quality :
 Class                  : 248
 Accuracy               : 254
 Offset (log variance) : 16640
Offset from master : 0 (ns)
Mean path delay    : 0 (ns)
Steps removed      : 0
Local clock time   : Sun Jan 15 20:57:29 2011
```

# Display brief PTP statistics on Device A.

```
[DeviceA] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
XGE1/0/1       Master        P2P              Two         0
```

# Display PTP clock information on Device B.

```
[DeviceB] display ptp clock
PTP profile         : IEEE 802.1AS
PTP mode            : P2PTC
Slave only          : No
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 246
Priority2     : 248
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 16640
Offset from master : N/A
Mean path delay    : N/A
Steps removed      : N/A
Local clock time   : Sun Jan 15 20:57:29 2011
```

# Display brief PTP statistics on Device B.

```
[DeviceB] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
XGE1/0/1       N/A           P2P              Two         0
XGE1/0/2       N/A           P2P              Two         0
```

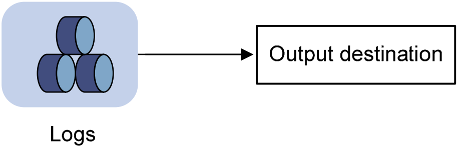
Configuring the information center

The information center on a device classifies and manages logs for all modules so that network administrators can monitor network performance and troubleshoot network problems.

Overview

The information center receives logs generated by source modules and outputs logs to different destinations according to user-defined output rules. You can classify, filter, and output logs based on source modules. To view the supported source modules, use **info-center source ?**.

Figure 23 Information center diagram



By default, the information center is enabled. It affects system performance to some degree while processing large amounts of information.

Log types

Logs are classified into the following types:

- **Common logs**—Record common system information. Unless otherwise specified, the term "logs" in this document refers to common logs.
- **Diagnostic logs**—Record debug messages.
- **Security logs**—Record security information, such as authentication and authorization information.
- **Hidden logs**—Record log information not displayed on the terminal, such as input commands.
- **Trace logs**—Record system tracing and debug messages, which can be viewed only after the devkit package is installed.

Log levels

Logs are classified into eight severity levels from 0 through 7 in descending order. The device outputs logs with a severity level that is higher than or equal to the specified level. For example, if you configure an output rule with a severity level of 6 (informational), logs that have a severity level from 0 to 6 are output.

Table 6 Log levels

Severity value	Level	Description
0	Emergency	The system is unusable. For example, the system authorization has expired.

Severity value	Level	Description
1	Alert	Action must be taken immediately. For example, traffic on an interface exceeds the upper limit.
2	Critical	Critical condition. For example, the device temperature exceeds the upper limit, the power module fails, or the fan tray fails.
3	Error	Error condition. For example, the link state changes.
4	Warning	Warning condition. For example, an interface is disconnected, or the memory resources are used up.
5	Notification	Normal but significant condition. For example, a terminal logs in to the device, or the device reboots.
6	Informational	Informational message. For example, a command or a ping operation is executed.
7	Debugging	Debug message.

Log destinations

The system outputs logs to the following destinations: console, monitor terminal, log buffer, log host, and log file. Log output destinations are independent and you can configure them after enabling the information center.

Default output rules for logs

A log output rule specifies the source modules and severity level of logs that can be output to a destination. Logs matching the output rule are output to the destination. Table 7 shows the default log output rules.

Table 7 Default output rules

Destination	Log source modules	Output switch	Severity
Console	All supported modules	Enabled	Debug
Monitor terminal	All supported modules	Disabled	Debug
Log host	All supported modules	Enabled	Informational
Log buffer	All supported modules	Enabled	Informational
Log file	All supported modules	Enabled	Informational

Default output rules for diagnostic logs

Diagnostic logs can only be output to the diagnostic log file, and cannot be filtered by source modules and severity levels. Table 8 shows the default output rule for diagnostic logs.

Table 8 Default output rule for diagnostic logs

Destination	Log source modules	Output switch	Severity
Diagnostic log file	All supported modules	Enabled	Debug

Default output rules for security logs

Security logs can only be output to the security log file, and cannot be filtered by source modules and severity levels. Table 9 shows the default output rule for security logs.

Table 9 Default output rule for security logs

Destination	Log source modules	Output switch	Severity
Security log file	All supported modules	Disabled	Debugging

Default output rules for hidden logs

Hidden logs can be output to the log host, the log buffer, and the log file. Table 10 shows the default output rules for hidden logs.

Table 10 Default output rules for hidden logs

Destination	Log source modules	Output switch	Severity
Log host	All supported modules	Enabled	Informational
Log buffer	All supported modules	Enabled	Informational
Log file	All supported modules	Enabled	Informational

Default output rules for trace logs

Trace logs can only be output to the trace log file, and cannot be filtered by source modules and severity levels. Table 11 shows the default output rules for trace logs.

Table 11 Default output rules for trace logs

Destination	Log source modules	Output switch	Severity
Trace log file	All supported modules	Enabled	Debugging

Log formats

The format of logs varies by output destination. Table 12 shows the original format of log information, which might be different from what you see. The actual format depends on the log resolution tool used.

Table 12 Log formats

Output destination	Format	Example
Console, monitor terminal, log buffer, or log file	Prefix Timestamp Sysname Module/Level/Mnemonic: Content	%Nov 24 14:21:43:502 2010 HP SYSLOG/6/SYSLOG_RESTART: System restarted -- HP Comware Software.
Log host	• Standard format: <PRI>Timestamp Sysname %%vvModule/Level/Mnemonic: Source; Content • unicom format: <PRI>Timestamp Hostip vvModule/Level/Serial_number: Content • cmcc format: <PRI>Timestamp Sysname %vvModule/Level/Mnemonic: Source Content	• Standard format: <190>Nov 24 16:22:21 2010 HP %%10SYSLOG/6/SYSLOG_RESTART: -DevIP=1.1.1.1; System restarted -- HP Comware Software. • unicom format: <189>Oct 13 16:48:08 2000 10.1.1.1 10IFNET/2/210231a64jx073000020: VTY logged in from 192.168.1.21 • cmcc format: <189>Oct 9 14:59:04 2009 Sysname %10SHELL/5/SHELL_LOGIN: VTY logged in from 192.168.1.21

Table 13 describes the fields in a log message.

Table 13 Log field description

Field	Description
Prefix (information type)	A log to a destination other than the log host has an identifier in front of the timestamp: • An identifier of percent sign (%) indicates a log with a level equal to or higher than informational. • An identifier of asterisk (*) indicates a debug log or a trace log. • An identifier of caret (^) indicates a diagnostic log.
PRI (priority)	A log destined to the log host has a priority identifier in front of the timestamp. The priority is calculated by using this formula: facility*8+level, where: • **facility** is the facility name. Facility names local0 through local7 correspond to values 16 through 23. The facility name can be configured with the **info-center loghost** command. It is used to identify log sources on the log host, and to query and filter the logs from specific log sources. • **level** ranges from 0 to 7. See Table 6 for more information about severity levels.
Timestamp	Records the time when the log was generated. Logs sent to the log host and those sent to the other destinations have different timestamp precisions, and their timestamp formats are configured with different commands. For more information, see Table 14 and Table 15.
Hostip	Source IP address of the log. If **info-center loghost source** is configured, this field displays the IP address of the specified source interface. Otherwise, this field displays the sysname. This field exists only in logs in unicom format that are sent to the log host.

Field	Description
Serial number	Serial number of the device that generated the log. This field exists only in logs in unicom format that are sent to the log host.
Sysname (host name or host IP address)	The sysname is the host name or IP address of the device that generated the log. You can use the **sysname** command to modify the name of the device.
%% (vendor ID)	Indicates that the information was generated by an HP device. This field exists only in logs sent to the log host.
vv (version information)	Identifies the version of the log, and has a value of 10. This field exists only in logs that are sent to the log host.
Module	Specifies the name of the module that generated the log. You can enter the **info-center source ?** command in system view to view the module list.
Level	Identifies the level of the log. See Table 6 for more information about severity levels.
Mnemonic	Describes the content of the log. It contains a string of up to 32 characters.
Source	Identifies the source of the log. It can take one of the following values: • IRF member ID. • IP address of the log sender.
Content	Provides the content of the log.

Table 14 Timestamp precisions and configuration commands

Item	Destined to the log host	Destined to the console, monitor terminal, log buffer, and log file
Precision	Seconds	Milliseconds
Command used to set the timestamp format	**info-center timestamp loghost**	**info-center timestamp**

Table 15 Description of the timestamp parameters

Timestamp parameters	Description	Example
boot	Time that has elapsed since system startup, in the format of xxx.yyy. xxx represents the higher 32 bits, and yyy represents the lower 32 bits, of milliseconds elapsed. Logs that are sent to all destinations other than a log host support this parameter.	%0.109391473 Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully. 0.109391473 is a timestamp in the **boot** format.
date	Current date and time, in the format of mmm dd hh:mm:ss yyy for logs that are output to a log host, or MMM DD hh:mm:ss:xxx YYYY for logs that are output to other destinations. All logs support this parameter.	%May 30 05:36:29:579 2003 Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully. May 30 05:36:29:579 2003 is a timestamp in the **date** format.

Timestamp parameters	Description	Example
iso	Timestamp format stipulated in ISO 8601. Only logs that are sent to a log host support this parameter.	<189>2003-05-30T06:42:44 Sysname %%10FTPD/5/FTPD_LOGIN(l): User ftp (192.168.1.23) has logged in successfully. 2003-05-30T06:42:44 is a timestamp in the **iso** format.
none	No timestamp is included. All logs support this parameter.	% Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully. No timestamp is included.
no-year-date	Current date and time without year information, in the format of MMM DD hh:mm:ss:xxx. Only logs that are sent to a log host support this parameter.	<189>May 30 06:44:22 Sysname %%10FTPD/5/FTPD_LOGIN(l): User ftp (192.168.1.23) has logged in successfully. May 30 06:44:22 is a timestamp in the **no-year-date** format.

FIPS compliance

The device supports the FIPS mode that complies with NIST FIPS 140-2 requirements. Support for features, commands, and parameters might differ in FIPS mode and non-FIPS mode. For more information about FIPS mode, see *Security Configuration Guide*.

Information center configuration task list

Task at a glance
Perform at least one of the following tasks: • Outputting logs to the console • Outputting logs to the monitor terminal • Outputting logs to a log host • Outputting logs to the log buffer • Saving logs to the log file
(Optional.) Managing security logs
(Optional.) Saving diagnostic logs to the diagnostic log file
(Optional.) Configuring the maximum size of the trace log file
(Optional.) Enabling synchronous information output
(Optional.) Enabling duplicate log suppression
(Optional.) Disabling an interface from generating link up or link down logs
(Optional.) Setting the minimum storage time for logs

Outputting logs to the console

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable the information center.	**info-center enable**	By default, the information center is enabled.
3.	Configure an output rule for the console.	**info-center source** { *module-name* | **default** } { **console** | **monitor** | **logbuffer** | **logfile** | **loghost** } { **deny** | **level** *severity* }	For information about default output rules, see "Default output rules for logs."
4.	(Optional.) Configure the timestamp format.	**info-center timestamp** { **boot** | **date** | **none** }	By default, the timestamp format is **date**.
5.	Return to user view.	**quit**	N/A
6.	(Optional.) Enable log output to the console.	**terminal monitor**	The default setting is enabled.
7.	Enable the display of debug information on the current terminal.	**terminal debugging**	By default, the display of debug information is disabled on the current terminal.
8.	(Optional.) Set the lowest severity level of logs that can be output to the console.	**terminal logging level** *severity*	The default setting is 6 (informational).

Outputting logs to the monitor terminal

Monitor terminals refer to terminals that log in to the device through the VTY line.

To output logs to the monitor terminal:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable the information center.	**info-center enable**	By default, the information center is enabled.
3.	Configure an output rule for the monitor terminal.	**info-center source** { *module-name* | **default** } { **console** | **monitor** | **logbuffer** | **logfile** | **loghost** } { **deny** | **level** *severity* }	For information about default output rules, see "Default output rules for logs."
4.	(Optional.) Configure the timestamp format.	**info-center timestamp** { **boot** | **date** | **none** }	By default, the timestamp format is **date**.
5.	Return to user view.	**quit**	N/A
6.	Enable log output to the monitor terminal.	**terminal monitor**	The default setting is enabled.
7.	Enable the display of debug information on the current terminal.	**terminal debugging**	By default, the display of debug information is disabled on the current terminal.

Step		Command	Remarks
8.	(Optional.) Set the lowest level of logs that can be output to the monitor terminal.	**terminal logging level** *severity*	The default setting is 6 (informational).

Outputting logs to a log host

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable the information center.	**info-center enable**	By default, the information center is enabled.
3.	Configure an output rule for outputting logs to a log host.	**info-center source** { *module-name* | **default** } { **console** | **monitor** | **logbuffer** | **logfile** | **loghost** } { **deny** | **level** *severity* }	For information about default output rules, see "Default output rules for logs."
4.	(Optional.) Specify the source IP address for output logs.	**info-center loghost source** *interface-type interface-number*	By default, the source IP address of output log information is the primary IP address of the matching route's egress interface.
5.	(Optional.) Specify the format for logs sent to a log host.	**info-center format** { **unicom** | **cmcc** }	By default, logs are sent in standard format to a log host.
6.	(Optional.) Configure the timestamp format.	**info-center timestamp loghost** { **date** | **iso** [**with-timezone**] | **no-year-date** | **none** }	By default, the timestamp format is **date**.
7.	Specify a log host and configure related parameters.	**info-center loghost** { *loghost* | *ipv4-address* | **ipv6** *ipv6-address* } [**port** *port-number*] [**facility** *local-number*]	By default, no log host or related parameters are specified. The value of the *port-number* argument must be the same as the value configured on the log host. Otherwise, the log host cannot receive logs.

Outputting logs to the log buffer

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable the information center.	**info-center enable**	By default, the information center is enabled.
3.	Enable log output to the log buffer.	**info-center logbuffer**	By default, log output to the log buffer is enabled.
4.	(Optional.) Set the maximum number of logs that can be stored in the log buffer.	**info-center logbuffer size** *buffersize*	By default, the log buffer can store 512 logs.

Step	Command	Remarks
5. Configure an output rule for the log buffer.	**info-center source** { *module-name* | **default** } { **console** | **monitor** | **logbuffer** | **logfile** | **loghost** } { **deny** | **level** *severity* }	For information about default output rules, see "Default output rules for logs."
6. (Optional.) Configure the timestamp format.	**info-center timestamp** { **boot** | **date** | **none** }	By default, the timestamp format is **date**.

Saving logs to the log file

Before logs are saved to the log file, they are output to a temporary buffer called the log file buffer. Do not confuse the log file buffer with log buffer, which is an independent log output destination.

By default, the log file feature saves logs from the log file buffer to the log file every 24 hours. You can adjust the saving interval or manually save logs to the log file. After saving logs to the log file, the system clears the log file buffer.

The log file has a maximum capacity. When the maximum capacity is reached, the system will replace earliest logs with new logs.

To save logs to the log file:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable the information center.	**info-center enable**	By default, the information center is enabled.
3. Enable the log file feature.	**info-center logfile enable**	By default, the log file feature is enabled.
4. (Optional.) Enable log file overwrite-protection.	**info-center logfile overwrite-protection** [**all-port-powerdown**]	By default, log file overwrite-protection is disabled. This command is supported only in FIPS mode.
5. (Optional.) Configure the maximum size for the log file.	**info-center logfile size-quota** *size*	By default, the maximum size of the log file is 10 MB. To ensure normal operation, set the *size* argument to a value between 1 MB and 10 MB.
6. (Optional.) Specify the directory to save the log file.	**info-center logfile directory** *dir-name*	The default directory is flash:/logfile. The configuration made by this command cannot survive an IRF reboot or a master/subordinate switchover.

Step	Command	Remarks
7. Save the logs in the log file buffer to the log file.	• Configure the interval to perform the save operation: **info-center logfile frequency** *freq-sec* • Manually save the logs in the log file buffer to the log file: **logfile save**	The default saving interval is 86400 seconds. Execute the **logfile save** command in any view.

Managing security logs

Security logs are very important for locating and troubleshooting network problems. Generally, security logs are output together with other logs. It is difficult to identify security logs among all logs.

To solve this problem, you can save security logs to the security log file without affecting the current log output rules.

Saving security logs to the security log file

After you enable the saving of the security logs to the security log file:

- The system first outputs security logs to the security log file buffer.
- The system saves the logs from the security log file buffer to the security log file at a specified interval (the security log administrator can also manually save security logs to the log file).
- After the security logs are saved, the buffer is cleared immediately.

The device supports only one security log file. To avoid security log loss, you can set an alarm threshold for the security log file usage. When the alarm threshold is reached, the system outputs a message to inform the administrator. The administrator can log in to the device as the security log administrator and back up the security log file to prevent the loss of important data.

To save security logs to the security log file:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable the information center.	**info-center enable**	By default, the information center is enabled.
3. Enable the saving of the security logs to the security log file.	**info-center security-logfile enable**	By default, saving security logs to the security log file is disabled.
4. Set the interval at which the system saves security logs.	**info-center security-logfile frequency** *freq-sec*	By default, the system saves security logs to the security log file every 86400 seconds.
5. (Optional.) Set the maximum size of the security log file.	**info-center security-logfile size-quota** *size*	By default, the maximum size of the security log file is 10 MB.

Step	Command	Remarks
6. (Optional.) Set the alarm threshold of the security log file usage.	**info-center security-logfile alarm-threshold** *usage*	By default, the alarm threshold of the security log file usage is 80. When the usage of the security log file reaches 80%, the system will inform the user.

Managing the security log file

To manage and maintain the security log file, the security log administrator must pass local AAA authentication first. For more information about security log administrator, see *Security Configuration Guide*.

To manage the security log file:

Task	Command	Remarks
Display a summary of the security log file.	**display security-logfile summary**	Available in user view.
Change the directory of the security log file.	1. **system-view** 2. **info-center security-logfile directory** *dir-name*	By default, the security log file is saved in the **seclog** directory in the root directory of the storage device. The configuration made by this command cannot survive an IRF reboot or a master/subordinate switchover.
Manually save all the contents in the security log file buffer to the security log file.	**security-logfile save**	Available in any view.

Saving diagnostic logs to the diagnostic log file

By default, the system saves diagnostic logs from the diagnostic log file buffer to the diagnostic log file every 24 hours. You can adjust the saving interval or manually save diagnostic logs to the diagnostic log file. After saving diagnostic logs to the diagnostic log file, the system clears the diagnostic log file buffer.

The diagnostic log file has a maximum capacity. When the capacity is reached, the system replaces earliest diagnostic logs with new logs.

To enable saving diagnostic logs to a diagnostic log file:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable the information center.	**info-center enable**	By default, the information center is enabled.
3. Enable saving diagnostic logs to the diagnostic log file.	**info-center logfile enable**	By default, saving diagnostic logs to the diagnostic log file is enabled.

Step	Command	Remarks
4. (Optional.) Configure the maximum size of the diagnostic log file.	**info-center diagnostic-logfile quota** *size*	By default, the maximum size is 10 MB. To ensure normal operation, set the *size* argument to a value between 1 MB and 10 MB.
5. (Optional.) Specify the directory to save diagnostic log files.	**info-center diagnostic-logfile directory** *dir-name*	The default directory is flash:/diagfile. This command cannot survive an IRF reboot or a master/subordinate switchover.
6. Save the diagnostic logs in the diagnostic log file buffer to the diagnostic log file.	• Configure the interval to perform the saving operation: **info-center diagnostic-logfile frequency** *freq-sec* • Manually save the diagnostic logs in the buffer to the diagnostic log file: **diagnostic-logfile save**	The default saving interval is 86400 seconds. The **diagnostic-logfile save** command is available in any view.

Configuring the maximum size of the trace log file

The device has only one trace log file. When the trace log file is full, the device overwrites the oldest trace logs with new ones.

To set the maximum size of the trace log file:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Set the maximum size of the trace log file.	**info-center diagnostic-logfile quota** *size*	By default, the maximum size of the trace log file is 1 MB.

Enabling synchronous information output

System log output interrupts ongoing configuration operations, obscuring previously entered commands. Synchronous information output shows the obscured commands. It also provides a command prompt in command editing mode, or a [Y/N] string in interaction mode so you can continue your operation from where you were stopped.

To enable synchronous information output:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step		Command	Remarks
2.	Enable synchronous information output.	**info-center synchronous**	By default, synchronous information output is disabled.

Enabling duplicate log suppression

The output of consecutive duplicate logs at an interval of less than 30 seconds wastes system and network resources.

With this feature enabled, the system starts a suppression period upon outputting a log:

- During the suppression period, the system does not output logs that have the same module name, level, mnemonic, location, and text as the previous log.
- After the suppression period expires, if the same log continues to appear, the system outputs the suppressed logs and the log number and starts another suppression period. The suppression period is 30 seconds for the first time, 2 minutes for the second time, and 10 minutes for subsequent times.
- If a different log is generated during the suppression period, the system aborts the current suppression period, outputs suppressed logs and the log number and then the different log, and starts another suppression period.

To enable duplicate log suppression:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable duplicate log suppression.	**info-center logging suppress duplicates**	By default, duplicate log suppression is disabled.

Disabling an interface from generating link up or link down logs

By default, all interfaces generate link up or link down log information when the interface state changes. In some cases, you might want to disable some interfaces from generating this information. For example:

- You are concerned only about the states of some interfaces. In this case, you can use this function to disable other interfaces from generating link up and link down log information.
- An interface is unstable and continuously outputs log information. In this case, you can disable the interface from generating link up and link down log information.

To disable an interface from generating link up or link down logs:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter interface view.	**interface** *interface-type interface-number*	N/A

Step	Command	Remarks
3. Disable the interface from generating link up or link down logs.	**undo enable log updown**	By default, all interfaces generate link up and link down logs when the interface state changes.

Setting the minimum storage time for logs

Use this feature to set the minimum storage time for logs in the log buffer and log file. A log will not be automatically deleted from the log buffer or log file if the time elapsed since the log was generated is shorter than the minimum storage time. By default, the system automatically deletes the earliest logs when the log buffer of log file is full, regardless of the log storage time.

To set the log minimum storage time:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Set the log minimum storage time.	**info-center syslog min-age** *min-age*	By default, the log minimum storage time is not configured.

Displaying and maintaining information center

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display the diagnostic log file configuration.	**display diagnostic-logfile summary**
Display the information of each output destination.	**display info-center**
Display the state and the log information of the log buffer.	**display logbuffer** [**reverse**] [**level** *severity* | **size** *buffersize* | **slot** *slot-number*] *
Display a summary of the log buffer.	**display logbuffer summary** [**level** *severity* **|** **slot** *slot-number*] *
Display the configuration of the log file.	**display logfile summary**
Clear the log buffer.	**reset logbuffer**

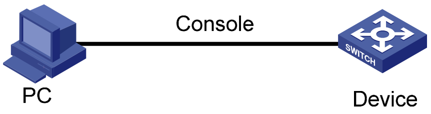
Information center configuration examples

Configuration example for outputting logs to the console

Network requirements

Configure the device to output to the console FTP logs that have a severity level of at least **warning**.

Figure 24 Network diagram



Configuration procedure

# Enable the information center.

```
<Device> system-view
[Device] info-center enable
```

# Disable log output to the console.

```
[Device] info-center source default console deny
```

To avoid output of unnecessary information, disable all modules from outputting log information to the specified destination (**console** in this example) before you configure the output rule.

# Configure an output rule to output to the console FTP logs that have a severity level of at least **warning**.

```
[Device] info-center source ftp console level warning
[Device] quit
```

# Enable the display of logs on the console. By default, the display of logs on the console is enabled.

```
<Device> terminal logging level 6
<Device> terminal monitor
 The current terminal is enabled to display logs.
```

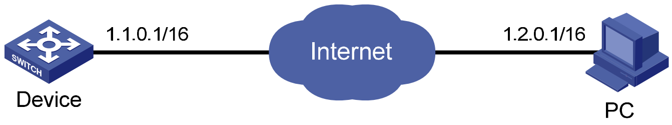
Now, if the FTP module generates logs, the information center automatically sends the logs to the console, and the console displays the logs.

Configuration example for outputting logs to a UNIX log host

Network requirements

Configure the device to output to the UNIX log host FTP logs that have a severity level of at least **informational**.

Figure 25 Network diagram



Configuration procedure

Before the configuration, make sure the device and the log host can reach each other. (Details not shown.)

1. Configure the device:

   # Enable the information center.

   ```
   <Device> system-view
   [Device] info-center enable
   ```

   # Specify the log host 1.2.0.1/16 and specify **local4** as the logging facility.

   ```
   [Device] info-center loghost 1.2.0.1 facility local4
   ```

# Disable log output to the log host.

```
[Device] info-center source default loghost deny
```

To avoid output of unnecessary information, disable all modules from outputting logs to the specified destination (**loghost** in this example) before you configure an output rule.

# Configure an output rule to output to the log host FTP logs that have a severity level of at least **informational**.

```
[Device] info-center source ftp loghost level informational
```

2. Configure the log host:

   The following configurations were performed on Solaris. Other UNIX operating systems have similar configurations.

   a. Log in to the log host as a root user.

   b. Create a subdirectory named **Device** in directory **/var/log/**, and then create file **info.log** in the **Device** directory to save logs from **Device**.

   ```
   # mkdir /var/log/Device
   # touch /var/log/Device/info.log
   ```

   c. Edit the file **syslog.conf** in directory **/etc/** and add the following contents.

   ```
   # Device configuration messages
   local4.info /var/log/Device/info.log
   ```

   In this configuration, **local4** is the name of the logging facility that the log host uses to receive logs. **info** is the informational level. The UNIX system records the log information that has a severity level of at least **informational** to the file **/var/log/Device/info.log**.

NOTE:

Follow these guidelines while editing the file **/etc/syslog.conf**:

- Comments must be on a separate line and must begin with a pound sign (#).
- No redundant spaces are allowed after the file name.
- The logging facility name and the severity level specified in the **/etc/syslog.conf** file must be identical to those configured on the device by using the **info-center loghost** and **info-center source** commands. Otherwise, the log information might not be output properly to the log host.

   d. Display the process ID of **syslogd**, kill the **syslogd** process, and then restart **syslogd** using the **–r** option to make the new configuration take effect.

   ```
   # ps -ae | grep syslogd
   147
   # kill -HUP 147
   # syslogd -r &
   ```

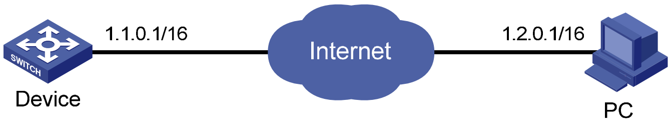
Now, the device can output FTP logs to the log host, which stores the logs to the specified file.

Configuration example for outputting logs to a Linux log host

Network requirements

Configure the device to output to the Linux log host 1.2.0.1/16 FTP logs that have a severity level of at least **informational**.

Figure 26 Network diagram



Configuration procedure

Before the configuration, make sure the device and the log host can reach each other. (Details not shown.)

1. Configure the device:

   # Enable the information center.

   ```
   <Device> system-view
   [Device] info-center enable
   ```

   # Specify the log host 1.2.0.1/16, and specify **local5** as the logging facility.

   ```
   [Device] info-center loghost 1.2.0.1 facility local5
   ```

   # Disable log output to the log host.

   ```
   [Device] info-center source default loghost deny
   ```

   To avoid outputting unnecessary information, disable all modules from outputting log information to the specified destination (**loghost** in this example) before you configure an output rule.

   # Configure an output rule to enable output to the log host FTP logs that have a severity level of at least **informational**.

   ```
   [Device] info-center source ftp loghost level informational
   ```

2. Configure the log host:

   The following configurations were performed on Solaris. Other UNIX operating systems have similar configurations.

   a. Log in to the log host as a root user.

   b. Create a subdirectory named **Device** in the directory **/var/log/**, and create file **info.log** in the **Device** directory to save logs of **Device**.

      ```
      # mkdir /var/log/Device
      # touch /var/log/Device/info.log
      ```

   c. Edit the file **syslog.conf** in directory **/etc/** and add the following contents.

      ```
      # Device configuration messages
      local5.info /var/log/Device/info.log
      ```

      In the above configuration, **local5** is the name of the logging facility used by the log host to receive logs. **info** is the informational level. The Linux system will store the log information with a severity level equal to or higher than **informational** to the file **/var/log/Device/info.log**.

NOTE:

Follow these guidelines while editing the file **/etc/syslog.conf**:

- Comments must be on a separate line and must begin with a pound sign (#).
- No redundant spaces are allowed after the file name.
- The logging facility name and the severity level specified in the **/etc/syslog.conf** file must be identical to those configured on the device by using the **info-center loghost** and **info-center source** commands. Otherwise, the log information might not be output properly to the log host.

d. Display the process ID of **syslogd**, kill the **syslogd** process, and then restart **syslogd** by using the **-r** option to apply the new configuration.

Make sure the **syslogd** process is started with the **-r** option on a Linux log host.

```
# ps -ae | grep syslogd
147
# kill -9 147
# syslogd -r &
```

Now, the system can record log information to the specified file.

Configuring SNMP

This chapter provides an overview of the Simple Network Management Protocol (SNMP) and guides you through the configuration procedure.

Overview

SNMP is an Internet standard protocol widely used for a management station to access and operate the devices on a network, regardless of their vendors, physical characteristics, and interconnect technologies.

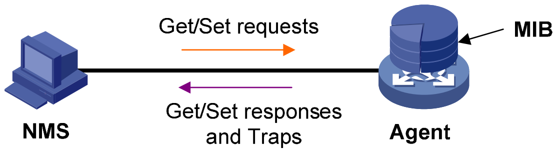
SNMP enables network administrators to read and set the variables on managed devices for state monitoring, troubleshooting, statistics collection, and other management purposes.

SNMP framework

The SNMP framework comprises the following elements:

- **SNMP manager**—Works on an NMS to monitor and manage the SNMP-capable devices in the network.
- **SNMP agent**—Works on a managed device to receive and handle requests from the NMS, and sends notifications to the NMS when events, such as an interface state change, occur.
- **Management Information Base (MIB)**—Specifies the variables (for example, interface status and CPU usage) maintained by the SNMP agent for the SNMP manager to read and set.

Figure 27 Relationship between NMS, agent, and MIB



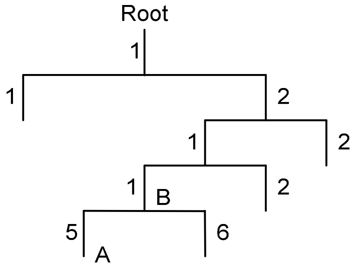
MIB and view-based MIB access control

A MIB stores variables called "nodes" or "objects" in a tree hierarchy and identifies each node with a unique OID. An OID is a dotted numeric string that uniquely identifies the path from the root node to a leaf node. For example, object B in Figure 28 is uniquely identified by the OID {1.2.1.1}.

For an NMS to identify the device models, each device model has a unique device OID (Devices of the same model use the same device OID.)

An IRF fabric does not have a device OID. When an NMS requests its device ID from an IRF fabric, the IRF fabric responds by using the OIDs of all its member devices.

Figure 28 MIB tree



A MIB view represents a set of MIB objects (or MIB object hierarchies) with certain access privileges and is identified by a view name. The MIB objects included in the MIB view are accessible while those excluded from the MIB view are inaccessible.

A MIB view can have multiple view records each identified by a *view-name oid-tree* pair.

You control access to the MIB by assigning MIB views to SNMP groups or communities.

For an NMS to identify the device model of each device, SNMP uses the device OID to uniquely identify a device. Devices of the same model share the same OID.

An IRF fabric does not have an independent OID. When an NMS requests the OID of an IRF fabric, the IRF fabric sends the OIDs of all IRF member devices.

SNMP operations

SNMP provides the following basic operations:

- **Get**—NMS retrieves the SNMP object nodes in an agent MIB.
- **Set**—NMS modifies the value of an object node in an agent MIB.
- **Notification**—SNMP agent sends traps or informs to report events to the NMS. The difference between these two types of notification is that informs require acknowledgment but traps do not. Traps are available in SNMPv1, SNMPv2c, and SNMPv3, but informs are available only in SNMPv2c and SNMPv3.

Protocol versions

SNMPv1, SNMPv2c, and SNMPv3 are supported in non-FIPS mode. Only SNMPv3 is supported in FIPS mode. An NMS and an SNMP agent must use the same SNMP version to communicate with each other.

- **SNMPv1**—Uses community names for authentication. To access an SNMP agent, an NMS must use the same community name as set on the SNMP agent. If the community name used by the NMS differs from the community name set on the agent, the NMS cannot establish an SNMP session to access the agent or receive traps from the agent.
- **SNMPv2c**—Uses community names for authentication. SNMPv2c is compatible with SNMPv1, but supports more operation types, data types, and error codes.
- **SNMPv3**—Uses a user-based security model (USM) to secure SNMP communication. You can configure authentication and privacy mechanisms to authenticate and encrypt SNMP packets for integrity, authenticity, and confidentiality.

Access control modes

SNMP uses the following modes to control access to MIB objects:

- **View-based Access Control Model**—The VACM mode controls access to MIB objects by assigning MIB views to SNMP communities or users.
- **Role based access control**—The RBAC mode controls access to MIB objects by assigning user roles to SNMP communities or users.
    - An SNMP community or user with a predefined user role network-admin or level-15 has read and write access to all MIB objects.
    - An SNMP community or user with a predefined user role network-operator has read-only access to all MIB objects.
    - An SNMP community or user with a user role specified by the **role** command accesses MIB objects through the user role rules specified by the **rule** command.

If you create the same SNMP community or user with both modes multiple times, the most recent configuration takes effect. For more information about user roles and the **rule** command, see *Fundamentals Command Reference*.

For an NMS to access an agent:

- The RBAC mode requires the user role bound to a community name or username to have the same access right to MIB objects as the NMS.
- The VACM mode requires only the access right from the NMS to MIB objects.

HP recommends the RBAC mode because it is more secure.

FIPS compliance

The device supports the FIPS mode that complies with NIST FIPS 140-2 requirements. Support for features, commands, and parameters might differ in FIPS mode and non-FIPS mode. For more information about FIPS mode, see *Security Configuration Guide*.

Configuring SNMP basic parameters

SNMPv3 differs from SNMPv1 and SNMPv2c in many ways. Their configuration procedures are described in separate sections.

Configuring SNMPv1 or SNMPv2c basic parameters

SNMPv1 and SNMPv2c settings are supported only in non-FIPS mode.

To configure SNMPv1 or SNMPv2c basic parameters:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. (Optional.) Enable the SNMP agent.	**snmp-agent**	By default, the SNMP agent is disabled. The SNMP agent is enabled when you use any command that begins with **snmp-agent** except for the **snmp-agent calculate-password** command.
3. (Optional.) Configure the system contact.	**snmp-agent sys-info contact** *sys-contact*	By default, the system contact is null.
4. (Optional.) Configure the system location.	**snmp-agent sys-info location** *sys-location*	By default, the system location is null.
5. Enable SNMPv1 or SNMPv2c.	**snmp-agent sys-info version** { **all** | { **v1** | **v2c** | **v3** } * }	By default, SNMPv3 is used.
6. (Optional.) Change the local engine ID.	**snmp-agent local-engineid** *engineid*	By default, the local engine ID is the company ID plus the device ID.
7. (Optional.) Create or update a MIB view.	**snmp-agent mib-view** { **excluded** | **included** } *view-name oid-tree* [**mask** *mask-value*]	By default, the MIB view **ViewDefault** is predefined. In this view, all the MIB objects in the **iso** subtree but the **snmpUsmMIB**, **snmpVacmMIB**, and **snmpModules.18** subtrees are accessible. Each *view-name oid-tree* pair represents a view record. If you specify the same record with different MIB sub-tree masks multiple times, the most recent configuration takes effect. Except for the four sub-trees in the default MIB view, you can create up to 16 unique MIB view records.

Step	Command	Remarks
8. Configure the SNMP access right.	- (Method 1.) Create an SNMP community: In VACM mode: **snmp-agent community** { **read** | **write** } [**simple** | **cipher**] *community-name* [**mib-view** *view-name*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * In RBAC mode: **snmp-agent community** [**simple** | **cipher**] *community-name* **user-role** *role-name* [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * - (Method 2.) Create an SNMPv1/v2c group, and add users to the group: a. **snmp-agent group** { **v1** | **v2c** } *group-name* [**read-view** *view-name*] [**write-view** *view-name*] [**notify-view** *view-name*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * b. **snmp-agent usm-user** { **v1** | **v2c** } *user-name group-name* [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] *	By default, no SNMP group or SNMP community exists. The username in method 2 has the same purpose as the community name in method 1. Whichever method you use, make sure the configured name is the same as the community name on the NMS.
9. (Optional.) Create an SNMP context.	**snmp-agent context** *context-name*	By default, no SNMP context is configured on the device.
10. (Optional.) Map an SNMP community to an SNMP context.	**snmp-agent community-map** *community-name* **context** *context-name*	By default, no mapping between an SNMP community and an SNMP context exists on the device.
11. (Optional.) Configure the maximum SNMP packet size (in bytes) that the SNMP agent can handle.	**snmp-agent packet max-size** *byte-count*	By default, the maximum SNMP packet size that the SNMP agent can handle is 1500 bytes. If the packet size of the requests and responses that contain MIB node information exceeds the maximum packet size that the agent can handle, operations from the NMS fail. For the NMS to access the agent successfully, configure a bigger packet size that the agent can handle.
12. Specify the UDP port for receiving SNMP packets.	**snmp-agent port** *port-number*	By default, the device uses UDP port 161 for receiving SNMP packets.

Configuring SNMPv3 basic parameters

SNMPv3 users are managed in groups. All SNMPv3 users in a group share the same security model, but can use different authentication and privacy key settings. To implement a security model for a user and avoid SNMP communication failures, make sure the security model configuration for the group and the security key settings for the user are compliant with Table 16 and match the settings on the NMS.

Table 16 Basic security setting requirements for different security models

Security model	Security model keyword for the group	Security key settings for the user	Remarks
Authentication with privacy	**privacy**	Authentication key, privacy key	If the authentication key or the privacy key is not configured, SNMP communication will fail.
Authentication without privacy	**authentication**	Authentication key	If no authentication key is configured, SNMP communication will fail. The privacy key (if any) for the user does not take effect.
No authentication, no privacy	Neither **authentication** nor **privacy**	None	The authentication and privacy keys, if configured, do not take effect.

To configure SNMPv3 basic parameters:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. (Optional.) Enable the SNMP agent.	**snmp-agent**	By default, the SNMP agent is disabled. The SNMP agent is enabled when you use any command that begins with **snmp-agent** except for the **snmp-agent calculate-password** command.
3. (Optional.) Configure the system contact.	**snmp-agent sys-info contact** *sys-contact*	By default, the system contact is null.
4. (Optional.) Configure the system location.	**snmp-agent sys-info location** *sys-location*	By default, the system location is null.
5. Enable SNMPv3.	• In non-FIPS mode: **snmp-agent sys-info version** { **all** | { **v1** | **v2c** | **v3** }* } • In FIPS mode: **snmp-agent sys-info version** { **all** | { **v1** | **v2c** | **v3** }* }	By default, SNMPv3 is used.

Step	Command	Remarks
6. (Optional.) Change the local engine ID.	**snmp-agent local-engineid** *engineid*	By default, the local engine ID is the company ID plus the device ID. IMPORTANT: After you change the local engine ID, the existing SNMPv3 users and encrypted keys become invalid, and you must reconfigure them.
7. (Optional.) Configure a remote engine ID.	**snmp-agent remote** { *ip-address* | **ipv6** *ipv6-address* } **engineid** *engineid*	By default, no remote engine ID is configured. To send informs to an SNMPv3 NMS, you must configure the SNMP engine ID of the NMS.
8. (Optional.) Create or update a MIB view.	**snmp-agent mib-view** { **excluded** | **included** } *view-name oid-tree* [**mask** *mask-value*]	By default, the MIB view **ViewDefault** is predefined. In this view, all the MIB objects in the **iso** subtree but the **snmpUsmMIB**, **snmpVacmMIB**, and **snmpModules.18** subtrees are accessible. Each *view-name oid-tree* pair represents a view record. If you specify the same record with different MIB sub-tree masks multiple times, the most recent configuration takes effect. Except for the four sub-trees in the default MIB view, you can create up to 16 unique MIB view records.
9. (Optional.) Create an SNMPv3 group.	• In non-FIPS mode: **snmp-agent group v3** *group-name* [**authentication** | **privacy**] [**read-view** *view-name*] [**write-view** *view-name*] [**notify-view** *view-name*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * • In FIPS mode: **snmp-agent group v3** *group-name* { **authentication** | **privacy** } [**read-view** *read-view*] [**write-view** *write-view*] [**notify-view** *notify-view*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] *	By default, no SNMP group exists.

Step	Command	Remarks

10. (Optional.) Calculate a digest for the ciphertext key converted from a plaintext key.	• In non-FIPS mode:
	snmp-agent calculate-password *plain-password* **mode** { **3desmd5** | **3dessha** | **md5** | **sha** } { **local-engineid** | **specified-engineid** *engineid* }
	• In FIPS mode:
	snmp-agent calculate-password *plain-password* **mode sha** { **local-engineid** | **specified-engineid** *engineid* }
		N/A

Step	Command	Remarks
11. Create an SNMPv3 user.	- In non-FIPS mode: In VACM mode: **snmp-agent usm-user v3** *user-name group-name* [**remote** { *ip-address* | **ipv6** *ipv6-address* }] [{ **cipher** | **simple** } **authentication-mode** { **md5** | **sha** } *auth-password* [**privacy-mode** { **aes128** | **3des** | **des56** } *priv-password*]] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * In RBAC mode: **snmp-agent usm-user v3** *user-name* **user-role** *role-name* [**remote** { *ip-address* | **ipv6** *ipv6-address* }] [{ **cipher** | **simple** } **authentication-mode** { **md5** | **sha** } *auth-password* [**privacy-mode** { **aes128** | **3des** | **des56** } *priv-password*]] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * - In FIPS mode: In VACM mode: **snmp-agent usm-user v3** *user-name group-name* [**remote** { *ip-address* | **ipv6** *ipv6-address* }] { **cipher** | **simple** } **authentication-mode** **sha** *auth-password* [**privacy-mode aes128** *priv-password*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * In RBAC mode: **snmp-agent usm-user v3** *user-name* **user-role** *role-name* [**remote** { *ip-address* | **ipv6** *ipv6-address* }] { **cipher** | **simple** } **authentication-mode** **sha** *auth-password* [**privacy-mode aes128** *priv-password*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] *	If the **cipher** keyword is specified, the arguments *auth-password* and *priv-password* are used as encrypted keys. To send informs to an SNMPv3 NMS, you must configure the **remote** *ip-address* option to specify the IP address of the NMS.
12. (Optional.) Create an SNMP context.	**snmp-agent context** *context-name*	By default, no SNMP context is configured on the device.

Step	Command	Remarks
13. (Optional.) Configure the maximum SNMP packet size (in bytes) that the SNMP agent can handle.	**snmp-agent packet max-size** *byte-count*	By default, the maximum SNMP packet size that the SNMP agent can handle is 1500 bytes. If the packet size of the requests and responses that contain MIB node information exceeds the maximum packet size that the agent can handle, operations from the NMS fail. For the NMS to access the agent successfully, configure a bigger packet size that the agent can handle.
14. (Optional.) Specify the UDP port for receiving SNMP packets.	**snmp-agent port** *port-number*	By default, the device uses UDP port 161 for receiving SNMP packets.

Configuring SNMP logging

The SNMP agent logs Get requests, Set requests, Set responses, and SNMP notifications, but does not log Get responses.

- **Get operation**—The agent logs the IP address of the NMS, name of the accessed node, and node OID.
- **Set operation**—The agent logs the NMS' IP address, name of accessed node, node OID, variable value, and error code and index for the Set operation.
- **Notification tracking**—The agent logs the SNMP notifications after sending them to the NMS.
- **Authentication failures from the NMS to the agent**—The agent logs the IP address of the NMS.

To configure SNMP logging:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. (Optional.) Enable SNMP logging.	**snmp-agent log** { **all** | **authfail** | **get-operation** | **set-operation** }	By default, SNMP logging is disabled.
3. (Optional.) Enable SNMP notification logging.	**snmp-agent trap log**	By default, SNMP notification logging is disabled.

Configuring SNMP notifications

The SNMP Agent sends notifications (traps and informs) to inform the NMS of significant events, such as link state changes and user logins or logouts. Unless otherwise stated, the **trap** keyword in the command line includes both traps and informs.

Enabling SNMP notifications

Enable an SNMP notification only if necessary. SNMP notifications are memory-intensive and might affect device performance.

To generate linkUp or linkDown notifications when the link state of an interface changes, you must perform the following tasks:

- Enable linkUp or linkDown notification globally by using the **snmp-agent trap enable standard** [**linkdown** | **linkup**] * command.
- Enable linkUp or linkDown notification on the interface by using the **enable snmp trap updown** command.

After you enable notifications for a module, whether the module generates notifications also depends on the configuration of the module. For more information, see the configuration guide for each module.

To enable SNMP notifications:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable notifications globally.	**snmp-agent trap enable** [**configuration** | *protocol* | **standard** [**authentication** | **coldstart** | **linkdown** | **linkup** | **warmstart**] * | **system**]	By default, SNMP configuration notifications, standard notifications, and system notifications are enabled. Whether other SNMP notifications are enabled varies by modules.
3.	Enter interface view.	**interface** *interface-type interface-number*	N/A
4.	Enable link state notifications.	**enable snmp trap updown**	By default, link state notifications are enabled.

Configuring the SNMP agent to send notifications to a host

You can configure the SNMP agent to send notifications as traps or informs to a host, typically an NMS, for analysis and management. Traps are less reliable and use fewer resources than informs, because an NMS does not send an acknowledgment when it receives a trap.

Configuration guidelines

When network congestion occurs or the destination is not reachable, the SNMP agent buffers notifications in a queue. You can configure the queue size and the notification lifetime (the maximum time that a notification can stay in the queue). A notification is deleted when its lifetime expires. When the notification queue is full, the oldest notifications are automatically deleted.

You can extend standard linkUp/linkDown notifications to include interface description and interface type, but must make sure that the NMS supports the extended SNMP messages.

To send informs, make sure:

- The SNMP agent and the NMS use SNMPv2c or SNMPv3.
- If SNMPv3 is used, you must configure the SNMP engine ID of the NMS when you configure SNMPv3 basic settings. Also, specify the IP address of the SNMP engine when you create the SNMPv3 user.

Configuration prerequisites

- Configure the SNMP agent with the same basic SNMP settings as the NMS. If SNMPv1 or SNMPv2c is used, you must configure a community name. If SNMPv3 is used, you must configure an SNMPv3 user, a MIB view, and a remote SNMP engine ID associated with the SNMPv3 user for notifications.
- The SNMP agent and the NMS can reach each other.

Configuration procedure

To configure the SNMP agent to send notifications to a host:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure a target host.	• Send traps to the target host: In non-FIPS mode: **snmp-agent target-host trap address udp-domain** { *target-host* | **ipv6** *target-host* } [**udp-port** *port-number*] **params securityname** *security-string* [**v1** | **v2c** | **v3** [**authentication** | **privacy**]] In FIPS mode: **snmp-agent target-host trap address udp-domain** { *target-host* | **ipv6** *target-host* } [**udp-port** *port-number*] **params securityname** *security-string* **v3** { **authentication** | **privacy** } • Send informs to the target host: In non-FIPS mode: **snmp-agent target-host inform address udp-domain** { *target-host* | **ipv6** *target-host* } [**udp-port** *port-number*] **params securityname** *security-string* { **v2c** | **v3** [**authentication** | **privacy**] } In FIPS mode: **snmp-agent target-host inform address udp-domain** { *target-host* | **ipv6** *target-host* } [**udp-port** *port-number*] **params securityname** *security-string* **v3** { **authentication** | **privacy** }	By default, no target host is configured.
3. (Optional.) Configure a source address for notifications.	**snmp-agent** { **inform** | **trap** } **source** *interface-type interface-number*	By default, SNMP uses the IP address of the outgoing routed interface as the source IP address.
4. (Optional.) Enable extended linkUp/linkDown notifications.	**snmp-agent trap if-mib link extended**	By default, the SNMP agent sends standard linkup/linkDown notifications.
5. (Optional.) Configure the notification queue size.	**snmp-agent trap queue-size** *size*	By default, the notification queue can hold 100 notification messages.

Step		Command	Remarks
6.	(Optional.) Configure the notification lifetime.	**snmp-agent trap life** *seconds*	The default notification lifetime is 120 seconds.
7.	(Optional.) Configure the interval for sending periodical notifications.	**snmp-agent trap periodical-interval** *interval-time*	The default is 60 seconds.

Displaying the SNMP settings

Execute **display** commands in any view. The **display snmp-agent community** command is supported only in non-FIPS mode.

Task	Command
Display SNMP agent system information, including the contact, physical location, and SNMP version.	**display snmp-agent sys-info** [**contact** | **location** | **version**] *
Display SNMP agent statistics.	**display snmp-agent statistics**
Display the local engine ID.	**display snmp-agent local-engineid**
Display SNMP group information.	**display snmp-agent group** [*group-name*]
Display remote engine IDs.	**display snmp-agent remote** [*ip-address* | **ipv6** *ipv6-address*]
Display basic information about the notification queue.	**display snmp-agent trap queue**
Display the modules that can generate notifications and their notification status (enable or disable).	**display snmp-agent trap-list**
Display SNMPv3 user information.	**display snmp-agent usm-user** [**engineid** *engineid* | **username** *user-name* | **group** *group-name*] *
Display SNMPv1 or SNMPv2c community information. (This command is not supported in FIPS mode.)	**display snmp-agent community** [**read** | **write**]
Display MIB view information.	**display snmp-agent mib-view** [**exclude** | **include** | **viewname** *view-name*]
Display SNMP MIB node information.	**display snmp-agent mib-node** [**details** | **index-node** | **trap-node** | **verbose**]
Display an SNMP context.	**display snmp-agent context** [*context-name*]

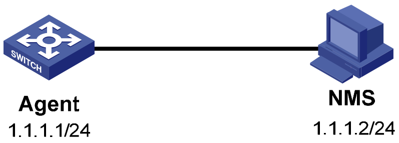
SNMPv1/SNMPv2c configuration example

SNMPv1 configuration procedure is the same as the SNMPv2c configuration procedure. This example uses SNMPv1, and is available only for non-FIPS mode.

Network requirements

As shown in Figure 29, the NMS (1.1.1.2/24) uses SNMPv1 to manage the SNMP agent (1.1.1.1/24), and the agent automatically sends notifications to report events to the NMS.

Figure 29 Network diagram



Configuration procedure

1. Configure the SNMP agent:

   # Configure the IP address of the agent and make sure the agent and the NMS can reach each other. (Details not shown.)

   # Specify SNMPv1, and create the read-only community **public** and the read and write community **private**.

   ```
   <Agent> system-view
   [Agent] snmp-agent sys-info version v1
   [Agent] snmp-agent community read public
   [Agent] snmp-agent community write private
   ```

   # Configure contact and physical location information for the agent.

   ```
   [Agent] snmp-agent sys-info contact Mr.Wang-Tel:3306
   [Agent] snmp-agent sys-info location telephone-closet,3rd-floor
   ```

   # Enable SNMP notifications, set the NMS at 1.1.1.2 as an SNMP trap destination, and use **public** as the community name. (To make sure the NMS can receive traps, specify the same SNMP version in the **snmp-agent target-host** command as is configured on the NMS.)

   ```
   [Agent] snmp-agent trap enable
   [Agent] snmp-agent target-host trap address udp-domain 1.1.1.2 params securityname
   public v1
   ```

2. Configure the SNMP NMS:
   - Specify SNMPv1.
   - Create the read-only community **public**, and create the read and write community **private**.
   - Configure timeout time and maximum number of retries as needed.

   For information about configuring the NMS, see the NMS manual.

NOTE:

The SNMP settings on the agent and the NMS must match.

3. Verify the configuration:

   # Try to get the MTU value of NULL0 interface from the agent. The attempt succeeds.

   ```
   Send request to 1.1.1.1/161 ...
   Protocol version: SNMPv1
   Operation: Get
   Request binding:
   1: 1.3.6.1.2.1.2.2.1.4.135471
   Response binding:
   1: Oid=ifMtu.135471 Syntax=INT Value=1500
   Get finished
   ```

   # Use a wrong community name to get the value of a MIB node on the agent. You can see an authentication failure trap on the NMS.

```
1.1.1.1/2934 V1 Trap = authenticationFailure
SNMP Version = V1
Community = public
Command = Trap
Enterprise = 1.3.6.1.4.1.43.1.16.4.3.50
GenericID = 4
SpecificID = 0
Time Stamp = 8:35:25.68
```

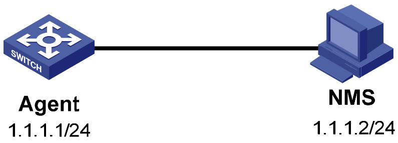
SNMPv3 in VACM mode configuration example

Network requirements

As shown in Figure 30, the NMS (1.1.1.2/24) uses SNMPv3 to monitor and manage the interface status of the agent (1.1.1.1/24). The agent automatically sends notifications to report events to the NMS.

The NMS and the agent perform authentication when they establish an SNMP session. The authentication algorithm is SHA-1 and the authentication key is **123456TESTauth&!**. The NMS and the agent also encrypt the SNMP packets between them by using the AES algorithm and the privacy key **123456TESTencr&!**.

Figure 30 Network diagram



Configuration procedure

1. Configure the agent:

   # Configure the IP address of the agent, and make sure the agent and the NMS can reach each other. (Details not shown.)

   # Assign the NMS (SNMPv3 group **managev3group**) read and write access to the objects under the **ifTable** node (OID 1.3.6.1.2.1.2.2), and deny its access to any other MIB object.

   ```
   <Agent> system-view
   [Agent] undo snmp-agent mib-view ViewDefault
   [Agent] snmp-agent mib-view included test ifTable
   [Agent] snmp-agent group v3 managev3group privacy read-view test write-view test
   ```

   # Add the user **managev3user** to the SNMPv3 group **managev3group**, and set the authentication algorithm to **sha**, authentication key to **123456TESTauth&!**, encryption algorithm to **aes128**, and privacy key to **123456TESTencr&!**.

   ```
   [Agent] snmp-agent usm-user v3 managev3user managev3group simple authentication-mode
   sha 123456TESTauth&! privacy-mode aes128 123456TESTencr&!
   ```

   # Configure contact and physical location information for the agent.

   ```
   [Agent] snmp-agent sys-info contact Mr.Wang-Tel:3306
   [Agent] snmp-agent sys-info location telephone-closet,3rd-floor
   ```

   # Enable notifications, specify the NMS at 1.1.1.2 as a trap destination, and set the username to **managev3user** for the traps.

   ```
   [Agent] snmp-agent trap enable
   ```

```
[Agent] snmp-agent target-host trap address udp-domain 1.1.1.2 params securityname
managev3user v3 privacy
```

2. Configure the SNMP NMS:
   - Specify SNMPv3.
   - Create the SNMPv3 user **managev3user**.
   - Enable both authentication and privacy functions.
   - Use SHA-1 for authentication and AES for encryption.
   - Set the authentication key to **123456TESTauth&!** and the privacy key to **123456TESTencr&!**.
   - Set the timeout time and maximum number of retries.

   For information about configuring the NMS, see the NMS manual.

NOTE:

The SNMP settings on the agent and the NMS must match.

3. Verify the configuration:

   # Try to get the MTU value of NULL0 interface from the agent. The get attempt succeeds.

```
Send request to 1.1.1.1/161 ...
Protocol version: SNMPv3
Operation: Get
Request binding:
1: 1.3.6.1.2.1.2.2.1.4.135471
Response binding:
1: Oid=ifMtu.135471 Syntax=INT Value=1500
Get finished
```

   # Try to get the device name from the agent. The get attempt fails because the NMS has no access right to the node.

```
Send request to 1.1.1.1/161 ...
Protocol version: SNMPv3
Operation: Get
Request binding:
1: 1.3.6.1.2.1.1.5.0
Response binding:
1: Oid=sysName.0 Syntax=noSuchObject Value=NULL
Get finished
```

   # Execute the **shutdown** or **undo shutdown** command on an idle interface on the agent. You can see the link state change traps on the NMS:

```
1.1.1.1/3374 V3 Trap = linkdown
SNMP Version = V3
Community = managev3user
Command = Trap
1.1.1.1/3374 V3 Trap = linkup
SNMP Version = V3
Community = managev3user
Command = Trap
```

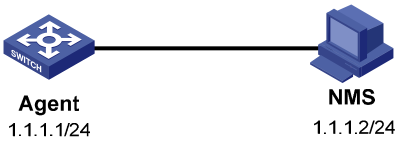
SNMPv3 in RBAC mode configuration example

Network requirements

As shown in Figure 31, the NMS (1.1.1.2/24) uses SNMPv3 to monitor and manage the interface status of the agent (1.1.1.1/24). The agent automatically sends notifications to report events to the NMS.

The NMS and the agent perform authentication when they establish an SNMP session. The authentication algorithm is SHA-1 and the authentication key is **123456TESTauth&!**. The NMS and the agent also encrypt the SNMP packets between them by using the AES algorithm and the privacy key **123456TESTencr&!**.

Figure 31 Network diagram



Configuration procedure

1. Configure the agent:

   # Configure the IP address of the agent, and make sure the agent and the NMS can reach each other. (Details not shown.)

   # Create the user role **test**, and permit **test** to have read and write access to the **snmp** node (OID 1.3.6.1.2.1.11).

```
<Agent> system-view
[Agent] role name test
[Agent-role-test] rule 1 permit read write oid 1.3.6.1.2.1.11
```

   # Permit the user role **test** to have read-only access to the **system** node (OID 1.3.6.1.2.1.1) and **hhpUIMgt** node (OID 1.3.6.1.4.1.25506.2.2).

```
[Agent-role-test] rule 2 permit read oid 1.3.6.1.4.1.25506.2.2
[Agent-role-test] rule 3 permit read oid 1.3.6.1.2.1.1
[Agent-role-test] quit
```

   # Create the SNMPv3 user **managev3user** with the user role **test**, and enable the authentication with privacy security model for the user. Set the authentication algorithm to **sha**, authentication key to **123456TESTauth&!**, encryption algorithm to **aes128**, and privacy key to **123456TESTencr&!**.

```
[Agent] snmp-agent usm-user v3 managev3user user-role test simple authentication-mode
sha 123456TESTauth&! privacy-mode aes128 123456TESTencr&!
```

   # Configure contact and physical location information for the agent.

```
[Agent] snmp-agent sys-info contact Mr.Wang-Tel:3306
[Agent] snmp-agent sys-info location telephone-closet,3rd-floor
```

   # Enable notifications, specify the NMS at 1.1.1.2 as a notification destination, and set the username to **managev3user** for the notifications.

```
[Agent] snmp-agent trap enable
[Agent] snmp-agent target-host trap address udp-domain 1.1.1.2 params securityname
managev3user v3 privacy
```

2. Configure the SNMP NMS:
   - Specify SNMPv3.

- Create the SNMPv3 user **managev3user**.
- Enable both authentication and privacy functions.
- Use SHA-1 for authentication and AES for encryption.
- Set the authentication key to **123456TESTauth&!** and the privacy key to **123456TESTencr&!**.
- Set the timeout time and maximum number of retries.

For information about configuring the NMS, see the NMS manual.

NOTE:

The SNMP settings on the agent and the NMS must match.

3. Verify the configuration:

# Try to get the value of **sysName** from the agent. The get attempt succeeds.

```
Send request to 1.1.1.1/161 ...
Protocol version: SNMPv3
Operation: Get
Request binding:
1: 1.3.6.1.2.1.1.5.0
Response binding:
1: Oid=sysName.0 Syntax=OCTETS Value=Agent
Get finished
```

# Try to set the device name from the agent. The set attempt fails because the NMS has no access right to the node.

```
Send request to 1.1.1.1/161 ...
Protocol version: SNMPv3
Operation: Set
Request binding:
1: 1.3.6.1.2.1.1.5.0
Response binding:
Session failed ! SNMP: Cannot access variable, No Access, error index=11:
Oid=sysName.0 Syntax=OCTETS Value=hp Set finished
%Aug 14 16:13:21:475 2013 Agent SNMP/5/SNMP_SETDENY:
-IPAddr=1.1.1.2-SecurityName=managev3user-SecurityModel=SNMPv3-OP=SET-Node=sysNam
e(1.3.6.1.2.1.1.5.0)-Value=hp; Permission denied.
```

# Log in to the agent. You can see a notification on the NMS.

```
hhpLogIn inform received from: 192.168.41.41 at 2013/8/14 17:36:16
  Time stamp: 0 days 08h:03m:43s.37th
  Agent address: 1.1.1.1 Port: 62861 Transport: IP/UDP Protocol: SNMPv2c Inform
  Manager address: 1.1.1.2 Port: 10005 Transport: IP/UDP
  Community: public
  Bindings (4)
    Binding #1: sysUpTime.0 *** (timeticks) 0 days 08h:03m:43s.37th
    Binding #2: snmpTrapOID.0 *** (oid) hhpLogIn
    Binding #3: hhpTerminalUserName.0 *** (octets) testuser [74.65.73.74.75.73.65.72
hex)]
    Binding #4: hhpTerminalSource.0 *** (octets) VTY [56.54.59 (hex)]
```

Configuring RMON

Overview

Remote Network Monitoring (RMON) is an enhancement to SNMP. It enables proactive remote monitoring and management of network devices and subnets. An RMON monitor periodically or continuously collects traffic statistics for the network attached to a port on the managed device. The managed device can automatically send a notification when a statistic crosses an alarm threshold, so the NMS does not need to constantly poll MIB variables and compare the results.

RMON uses SNMP notifications to notify NMSs of various alarm conditions such as broadcast traffic threshold exceeded. In contrast, SNMP reports function and interface operating status changes such as link up, link down, and module failure. For more information about SNMP notifications, see "Configuring SNMP"

HP devices provide an embedded RMON agent as the RMON monitor. An NMS can perform basic SNMP operations to access the RMON MIB.

RMON groups

Among standard RMON groups, HP implements the statistics group, history group, event group, alarm group, probe configuration group, and user history group. HP also implements a private alarm group, which enhances the standard alarm group. The probe configuration group and user history group are not configurable from the CLI. To configure these two groups, you must access the MIB.

Statistics group

The statistics group samples traffic statistics for monitored Ethernet interfaces and stores the statistics in the Ethernet statistics table (ethernetStatsTable). The statistics include:

- Number of collisions.
- CRC alignment errors.
- Number of undersize or oversize packets.
- Number of broadcasts.
- Number of multicasts.
- Number of bytes received.
- Number of packets received.

The statistics in the Ethernet statistics table are cumulative sums.

History group

The history group periodically samples traffic statistics on interfaces and saves the history samples in the history table (etherHistoryTable). The statistics include:

- Bandwidth utilization.
- Number of error packets.
- Total number of packets.

The history table stores traffic statistics collected for each sampling interval.

Event group

The event group controls the generation and notifications of events triggered by the alarms defined in the alarm group and the private alarm group. The following are RMON alarm event handling methods:

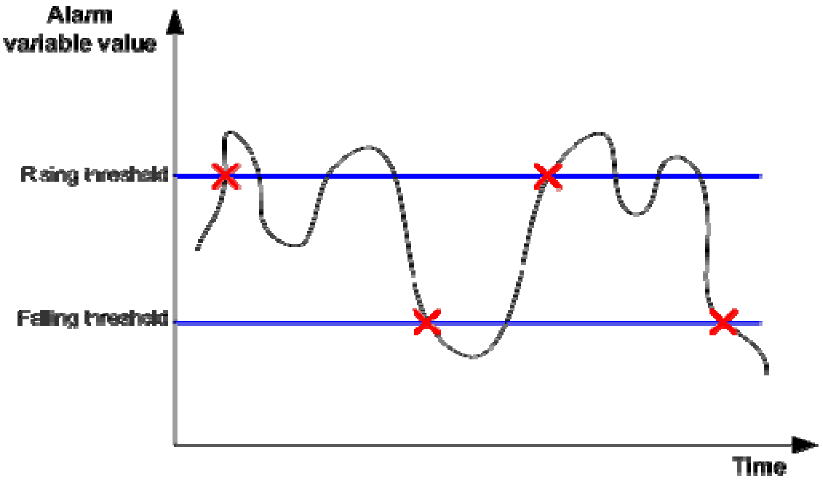
- **Log**—Logs event information (including event time and description) in the event log table so the management device can get the logs through SNMP.
- **Trap**—Sends an SNMP notification when the event occurs.
- **Log-Trap**—Logs event information in the event log table and sends an SNMP notification when the event occurs.
- **None**—Takes no actions.

Alarm group

The RMON alarm group monitors alarm variables, such as the count of incoming packets (etherStatsPkts) on an interface. After you create an alarm entry, the RMON agent samples the value of the monitored alarm variable regularly. If the value of the monitored variable is greater than or equal to the rising threshold, a rising alarm event is triggered. If the value of the monitored variable is smaller than or equal to the falling threshold, a falling alarm event is triggered. The event group defines the action to take on the alarm event.

If an alarm entry crosses a threshold multiple times in succession, the RMON agent generates an alarm event only for the first crossing. For example, if the value of a sampled alarm variable crosses the rising threshold multiple times before it crosses the falling threshold, only the first crossing triggers a rising alarm event, as shown in Figure 32.

Figure 32 Rising and falling alarm events



Private alarm group

The private alarm group enables you to perform basic math operations on multiple variables, and compare the calculation result with the rising and falling thresholds.

The RMON agent samples variables and takes an alarm action based on a private alarm entry as follows:

1. Samples the private alarm variables in the user-defined formula.
2. Processes the sampled values with the formula.

3. Compares the calculation result with the predefined thresholds, and then takes one of the following actions:
    - Triggers the event associated with the rising alarm event if the result is equal to or greater than the rising threshold.
    - Triggers the event associated with the falling alarm event if the result is equal to or less than the falling threshold.

If a private alarm entry crosses a threshold multiple times in succession, the RMON agent generates an alarm event only for the first crossing. For example, if the value of a sampled alarm variable crosses the rising threshold multiple times before it crosses the falling threshold, only the first crossing triggers a rising alarm event.

Sample types for the alarm group and the private alarm group

The RMON agent supports the following sample types:

- **absolute**—RMON compares the value of the monitored variable with the rising and falling thresholds at the end of the sampling interval.
- **delta**—RMON subtracts the value of the monitored variable at the previous sample from the current value, and then compares the difference with the rising and falling thresholds.

Protocols and standards

- RFC 4502, *Remote Network Monitoring Management Information Base Version 2*
- RFC 2819, *Remote Network Monitoring Management Information Base Status of this Memo*

Configuring the RMON statistics function

RMON implements the statistics function through the Ethernet statistics group and the history group. The statistics function is available only for Layer 2 Ethernet interfaces.

The Ethernet statistics group provides the cumulative statistic for a variable from the time the statistics entry is created to the current time. For more information about the configuration, see "Creating an RMON Ethernet statistics entry."

The history group provides statistics that are sampled for a variable for each sampling interval. The history group uses the history control table to control sampling, and it stores samples in the history table. For more information about the configuration, see "Creating an RMON history control entry."

Creating an RMON Ethernet statistics entry

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter Ethernet interface view.	**interface** *interface-type interface-number*	N/A

Step		Command	Remarks
3.	Create an entry for the interface in the RMON statistics table.	**rmon statistics** *entry-number* [**owner** *text*]	By default, the RMON statistics table does not contain entries. You can create one statistics entry for each Ethernet interface, and up to 100 statistics entries on the device. After the entry limit is reached, you cannot add new entries.

Creating an RMON history control entry

You can configure multiple history control entries for one interface, but you must make sure their entry numbers and sampling intervals are different.

You can create a history control entry successfully even if the specified bucket size exceeds the available history table size. RMON will set the bucket size as closely to the expected bucket size as possible.

To create an RMON history control entry:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3.	Create an entry for the interface in the RMON history control table.	**rmon history** *entry-number* **buckets** *number* **interval** *sampling-interval* [**owner** *text*]	By default, the RMON history control table does not contain entries. You can create up to 100 history control entries.

Configuring the RMON alarm function

When you configure the RMON alarm function, follow these guidelines:

- To send notifications to the NMS when an alarm is triggered, configure the SNMP agent as described in "Configuring SNMP" before configuring the RMON alarm function.
- For a new event, alarm, or private alarm entry to be created:
    - The entry must not have the same set of parameters as an existing entry.
    - The maximum number of entries is not reached.

Table 17 shows the parameters to be compared for duplication and the entry limits.

Table 17 RMON configuration restrictions

Entry	Parameters to be compared	Maximum number of entries
Event	• Event description (**description** *string*) • Event type (**log**, **trap**, **logtrap**, or **none**) • Community name (*security-string*)	60

Entry	Parameters to be compared	Maximum number of entries
Alarm	• Alarm variable (*alarm-variable*) • Sampling interval (*sampling-interval*) • Sample type (**absolute** or **delta**) • Rising threshold (*threshold-value1*) • Falling threshold (*threshold-value2*)	60
Private alarm	• Alarm variable formula (*prialarm-formula*) • Sampling interval (*sampling-interval*) • Sample type (**absolute** or **delta**) • Rising threshold (*threshold-value1*) • Falling threshold (*threshold-value2*)	50

To configure the RMON alarm function:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. (Optional.) Create an event entry in the event table.	**rmon event** *entry-number* [**description** *string*] { **log** | **log-trap** *security-string* | **none** | **trap** *security-string* } [**owner** *text*]	By default, the RMON event table does not contain entries.
3. Create an entry in the alarm table or private alarm table.	• Create an entry in the alarm table: **rmon alarm** *entry-number alarm-variable sampling-interval* { **absolute** | **delta** } [**startup-alarm** { **falling** | **rising** | **rising-falling** }] **rising-threshold** *threshold-value1 event-entry1* **falling-threshold** *threshold-value2 event-entry2* [**owner** *text*] • Create an entry in the private alarm table: **rmon prialarm** *entry-number prialarm-formula prialarm-des sampling-interval* { **absolute** | **delta** } [**startup-alarm** { **falling** | **rising** | **rising-falling** }] **rising-threshold** *threshold-value1 event-entry1* **falling-threshold** *threshold-value2 event-entry2* **entrytype** { **forever** | **cycle** *cycle-period* } [**owner** *text*]	By default, the RMON alarm table and the private alarm table do not contain entries. You can associate an alarm with an event that has not been created yet, but the alarm will trigger the event only after the event is created.

Displaying and maintaining RMON settings

Execute **display** commands in any view.

Task	Command
Display RMON statistics.	**display rmon statistics** [*interface-type interface-number*]
Display RMON history control entries and history samples.	**display rmon history** [*interface-type interface-number*]
Display RMON alarm entries.	**display rmon alarm** [*entry-number*]

Task	Command
Display RMON private alarm entries.	**display rmon prialarm** [*entry-number*]
Display RMON event entries.	**display rmon event** [*entry-number*]
Display log information for event entries.	**display rmon eventlog** [*entry-number*]

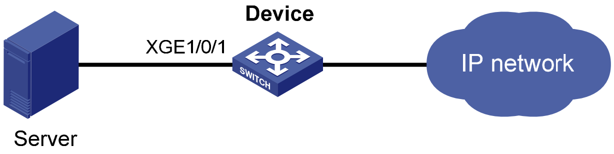
RMON configuration examples

Ethernet statistics group configuration example

Network requirements

As shown in Figure 33, create an RMON Ethernet statistics entry on the device to gather cumulative traffic statistics for Ten-GigabitEthernet 1/0/1.

Figure 33 Network diagram



Configuration procedure

# Create an RMON Ethernet statistics entry for Ten-GigabitEthernet 1/0/1.

```
<Sysname> system-view
[Sysname] interface ten-gigabitethernet 1/0/1
[Sysname-Ten-GigabitEthernet1/0/1] rmon statistics 1 owner user1
```

# Display statistics collected by the RMON agent for Ten-GigabitEthernet 1/0/1.

```
<Sysname> display rmon statistics ten-gigabitethernet 1/0/1
EtherStatsEntry 1 owned by user1 is VALID.
  Interface : Ten-GigabitEthernet1/0/1<ifIndex.3>
  etherStatsOctets         : 21657     , etherStatsPkts          : 307
  etherStatsBroadcastPkts  : 56        , etherStatsMulticastPkts : 34
  etherStatsUndersizePkts  : 0         , etherStatsOversizePkts  : 0
  etherStatsFragments      : 0         , etherStatsJabbers       : 0
  etherStatsCRCAlignErrors : 0         , etherStatsCollisions    : 0
  etherStatsDropEvents (insufficient resources): 0
  Incoming packets by size:
  64     : 235       ,  65-127  : 67        ,  128-255  : 4
  256-511: 1         ,  512-1023: 0         ,  1024-1518: 0
```

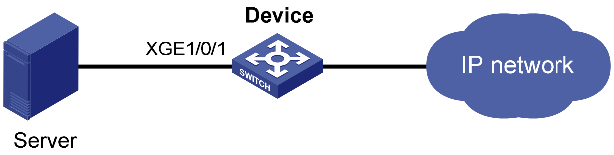
# Get the traffic statistics from the NMS through SNMP. (Details not shown.)

History group configuration example

Network requirements

As shown in Figure 34, create an RMON history control entry on the device to sample traffic statistics for Ten-GigabitEthernet 1/0/1 every minute.

Figure 34 Network diagram



Configuration procedure

# Create an RMON history control entry to sample traffic statistics every minute for Ten-GigabitEthernet 1/0/1. Retain up to eight samples for the interface in the history statistics table.

```
<Sysname> system-view
[Sysname] interface ten-gigabitethernet 1/0/1
[Sysname-Ten-GigabitEthernet1/0/1] rmon history 1 buckets 8 interval 60 owner user1
```

# Display the history statistics collected for Ten-GigabitEthernet 1/0/1.

```
[Sysname-Ten-GigabitEthernet1/0/1] display rmon history
HistoryControlEntry 1 owned by user1 is VALID
  Sampled interface     : Ten-GigabitEthernet1/0/1<ifIndex.3>
  Sampling interval     : 60(sec) with 8 buckets max
  Sampling record 1 :
    dropevents        : 0         , octets               : 834
    packets           : 8         , broadcast packets    : 1
    multicast packets : 6         , CRC alignment errors : 0
    undersize packets : 0         , oversize packets     : 0
    fragments         : 0         , jabbers              : 0
    collisions        : 0         , utilization          : 0
  Sampling record 2 :
    dropevents        : 0         , octets               : 962
    packets           : 10        , broadcast packets    : 3
    multicast packets : 6         , CRC alignment errors : 0
    undersize packets : 0         , oversize packets     : 0
    fragments         : 0         , jabbers              : 0
    collisions        : 0         , utilization          : 0
  Sampling record 3 :
    dropevents        : 0         , octets               : 830
    packets           : 8         , broadcast packets    : 0
    multicast packets : 6         , CRC alignment errors : 0
    undersize packets : 0         , oversize packets     : 0
    fragments         : 0         , jabbers              : 0
    collisions        : 0         , utilization          : 0
  Sampling record 4 :
    dropevents        : 0         , octets               : 933
    packets           : 8          , broadcast packets    : 0
    multicast packets : 7          , CRC alignment errors : 0
    undersize packets : 0          , oversize packets     : 0
    fragments         : 0          , jabbers              : 0
    collisions        : 0          , utilization          : 0
  Sampling record 5 :
    dropevents        : 0          , octets               : 898
    packets           : 9          , broadcast packets    : 2
    multicast packets : 6          , CRC alignment errors : 0
    undersize packets : 0          , oversize packets     : 0
    fragments         : 0          , jabbers              : 0
    collisions        : 0          , utilization          : 0
  Sampling record 6 :
    dropevents        : 0          , octets               : 898
    packets           : 9          , broadcast packets    : 2
    multicast packets : 6          , CRC alignment errors : 0
    undersize packets : 0          , oversize packets     : 0
    fragments         : 0          , jabbers              : 0
    collisions        : 0          , utilization          : 0
  Sampling record 7 :
    dropevents        : 0          , octets               : 766
    packets           : 7          , broadcast packets    : 0
    multicast packets : 6          , CRC alignment errors : 0
    undersize packets : 0          , oversize packets     : 0
    fragments         : 0          , jabbers              : 0
    collisions        : 0          , utilization          : 0
  Sampling record 8 :
    dropevents        : 0          , octets               : 1154
    packets           : 13         , broadcast packets    : 1
    multicast packets : 6          , CRC alignment errors : 0
    undersize packets : 0          , oversize packets     : 0
    fragments         : 0          , jabbers              : 0
    collisions        : 0          , utilization          : 0
```

# Get the traffic statistics from the NMS through SNMP. (Details not shown.)

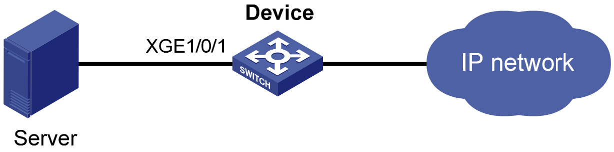
Alarm function configuration example

Network requirements

As shown in Figure 35, configure the device to monitor the incoming traffic statistic on Ten-GigabitEthernet 1/0/1, and send RMON alarms when the following events occur:

- The 5-second delta sample for the traffic statistic crosses the rising threshold (100).
- The 5-second delta sample for the traffic statistic drops below the falling threshold (50).

Figure 35 Network diagram



Configuration procedure

# Configure the SNMP agent (the device) with the same SNMP settings as the NMS at 1.1.1.2. This example uses SNMPv1, read community **public**, and write community **private**.

```
<Sysname> system-view
[Sysname] snmp-agent
[Sysname] snmp-agent community read public
[Sysname] snmp-agent community write private
[Sysname] snmp-agent sys-info version v1
[Sysname] snmp-agent trap enable
[Sysname] snmp-agent trap log
[Sysname] snmp-agent target-host trap address udp-domain 1.1.1.2 params securityname
public
```

# Create an RMON Ethernet statistics entry for Ten-GigabitEthernet 1/0/1.

```
[Sysname] interface ten-gigabitethernet 1/0/1
[Sysname-Ten-GigabitEthernet1/0/1] rmon statistics 1 owner user1
[Sysname-Ten-GigabitEthernet1/0/1] quit
```

# Create an RMON event entry and an RMON alarm entry to send SNMP notifications when the delta sample for 1.3.6.1.2.1.16.1.1.1.4.1 exceeds 100 or drops below 50.

```
[Sysname] rmon event 1 trap public owner user1
[Sysname] rmon alarm 1 1.3.6.1.2.1.16.1.1.1.4.1 5 delta rising-threshold 100 1
falling-threshold 50 1 owner user1
```

NOTE:

1.3.6.1.2.1.16.1.1.1.4.1 is the object instance for Ten-GigabitEthernet 1/0/1, where 1.3.6.1.2.1.16.1.1.1.4 represents the object for total incoming traffic statistics, and 1 is the RMON Ethernet statistics entry index for Ten-GigabitEthernet 1/0/1.

# Display the RMON alarm entry.

```
<Sysname> display rmon alarm 1
AlarmEntry 1 owned by user1 is VALID.
  Sample type          : delta
  Sampled variable     : 1.3.6.1.2.1.16.1.1.1.4.1<etherStatsOctets.1>
  Sampling interval (in seconds)    : 5
  Rising threshold      : 100(associated with event 1)
  Falling threshold     : 50(associated with event 1)
  Alarm sent upon entry startup  : risingOrFallingAlarm
  Latest value          : 0
```

# Display statistics for Ten-GigabitEthernet 1/0/1.

```
<Sysname> display rmon statistics ten-gigabitethernet 1/0/1
EtherStatsEntry 1 owned by user1 is VALID.
```

```
Interface : Ten-GigabitEthernet1/0/1<ifIndex.3>
etherStatsOctets         : 57329     , etherStatsPkts          : 455
etherStatsBroadcastPkts  : 53        , etherStatsMulticastPkts : 353
etherStatsUndersizePkts  : 0         , etherStatsOversizePkts  : 0
etherStatsFragments      : 0         , etherStatsJabbers       : 0
etherStatsCRCAlignErrors : 0         , etherStatsCollisions    : 0
etherStatsDropEvents (insufficient resources): 0
Incoming packets by size :
64     : 7         ,  65-127  : 413       ,  128-255  : 35
256-511: 0         ,  512-1023: 0         ,  1024-1518: 0
```

# Query alarm events on the NMS. (Details not shown.)

On the device, alarm event messages are displayed when events occur.

Configuring NQA

Overview

Network quality analyzer (NQA) allows you to measure network performance, verify the service levels for IP services and applications, and troubleshoot network problems. It provides the following types of operations:

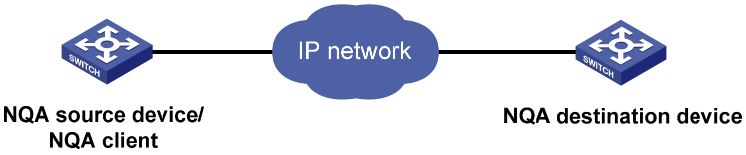
- ICMP echo.
- DHCP.
- DNS.
- FTP.
- HTTP.
- UDP jitter.
- SNMP.
- TCP.
- UDP echo.
- Voice.
- Path jitter.
- DLSw.

As shown in Figure 36, the NQA source device (NQA client) sends data to the NQA destination device by simulating IP services and applications to measure network performance. The obtained performance metrics include the one-way latency, jitter, packet loss, voice quality, application performance, and server response time.

All types of NQA operations require the NQA client, but only the TCP, UDP echo, UDP jitter, and voice operations require the NQA server. The NQA operations for services that are already provided by the destination device such as FTP do not need the NQA server.

You can configure the NQA server to listen and respond to specific IP addresses and ports to meet various test needs.

Figure 36 Network diagram



NQA operation

The following describes how NQA performs different types of operations:

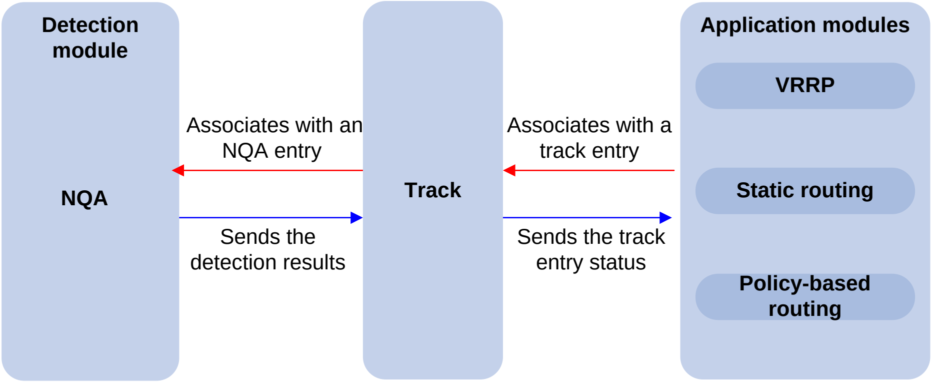
- A TCP or DLSw operation sets up a connection.

- A UDP jitter or a voice operation sends a number of probe packets. The number of probe packets is set by using the **probe packet-number** command.
- An FTP operation uploads or downloads a file.
- An HTTP operation gets a Web page.
- A DHCP operation gets an IP address through DHCP.
- A DNS operation translates a domain name to an IP address.
- An ICMP echo operation sends an ICMP echo request.
- A UDP echo operation sends a UDP packet.
- An SNMP operation sends one SNMPv1 packet, one SNMPv2c packet, and one SNMPv3 packet.
- A path jitter operation is accomplished in the following steps:
  a. The operation uses tracert to obtain the path from the NQA client to the destination. At maximum of 64 hops can be detected.
  b. The NQA client sends ICMP echo requests to each hop along the path. The number of ICMP echo requests is set by using the **probe packet-number** command.

Collaboration

NQA can collaborate with the Track module to notify application modules of state or performance changes so that the application modules can take predefined actions.

Figure 37 Collaboration


Detection module
NQA
Associates with an NQA entry
Sends the detection results
Track
Associates with a track entry
Sends the track entry status
Application modules
VRRP
Static routing
Policy-based routing

The following describes how a static route destined for 192.168.0.88 is monitored through collaboration:

1. NQA monitors the reachability to 192.168.0.88.
2. When 192.168.0.88 becomes unreachable, NQA notifies the Track module of the change.
3. The Track module notifies the static routing module of the state change.
4. The static routing module sets the static route as invalid according to a predefined action.

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

Threshold monitoring

Threshold monitoring enables the NQA client to take a predefined action when the NQA operation performance metrics violate the specified thresholds.

Table 18 describes the relationships between performance metrics and NQA operation types.

Table 18 Performance metrics and NQA operation types

Performance metric	NQA operation types that can gather the metric
Probe duration	All NQA operation types except UDP jitter, path jitter, and voice
Number of probe failures	All NQA operation types except UDP jitter, path jitter, and voice
Round-trip time	UDP jitter and voice
Number of discarded packets	UDP jitter and voice
One-way jitter (source-to-destination or destination-to-source)	UDP jitter and voice
One-way delay (source-to-destination or destination-to-source)	UDP jitter and voice
Calculated Planning Impairment Factor (ICPIF) (see "Configuring the voice operation")	Voice
Mean Opinion Scores (MOS) (see "Configuring the voice operation")	Voice

NQA configuration task list

Tasks at a glance	Remarks
Configuring the NQA server	Required for TCP, UDP echo, UDP jitter, and voice operations.
(Required.) Enabling the NQA client	N/A
(Required.) Perform at least one of the following tasks: • Configuring NQA operations on the NQA client • Configuring NQA templates on the NQA client	When you configure an NQA template to analyze network performance, the feature that uses the template performs the NQA operation.

Configuring the NQA server

To perform TCP, UDP echo, UDP jitter, and voice operations, you must enable the NQA server on the destination device. The NQA server listens and responds to requests on the specified IP addresses and ports.

You can configure multiple TCP or UDP listening services on an NQA server, where each corresponds to a specific IP address and port number. The IP address and port number for a listening service must be unique on the NQA server and match the configuration on the NQA client.

To configure the NQA server:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. Enable the NQA server.	**nqa server enable**	By default, the NQA server is disabled.
3. Configure a TCP or UDP listening service.	• TCP listening service: **nqa server tcp-connect** *ip-address* *port-number* [**tos** *tos*] • UDP listening service: **nqa server udp-echo** *ip-address* *port-number* [**tos** *tos*]	You can set the ToS value in the IP header of reply packets sent by the NQA server. The default ToS value is 0.

Enabling the NQA client

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable the NQA client.	**nqa agent enable**	By default, the NQA client is enabled.

Configuring NQA operations on the NQA client

NQA operation configuration task list

Tasks at a glance
(Required.) Perform at least one of the following tasks: • Configuring the ICMP echo operation • Configuring the DHCP operation • Configuring the DNS operation • Configuring the FTP operation • Configuring the HTTP operation • Configuring the UDP jitter operation • Configuring the SNMP operation • Configuring the TCP operation • Configuring the UDP echo operation • Configuring the voice operation • Configuring the DLSw operation • Configuring the path jitter operation
(Optional.) Configuring optional parameters for the NQA operation
(Optional.) Configuring the collaboration function
(Optional.) Configuring threshold monitoring
(Optional.) Configuring the NQA statistics collection function
(Optional.) Configuring the saving of NQA history records
(Required.) Scheduling the NQA operation on the NQA client

Configuring the ICMP echo operation

The ICMP echo operation measures the reachability of a destination device. It has the same function as the **ping** command, but provides more output information. In addition, if multiple paths exist between the source and destination devices, you can specify the next hop for the ICMP echo operation.

The ICMP echo operation is not supported in IPv6 networks. To test the reachability of an IPv6 address, use the **ping ipv6** command. For more information about the command, see *Network Management and Monitoring Command Reference*.

To configure the ICMP echo operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify the ICMP echo type and enter its view.	**type icmp-echo**	N/A
4. Specify the destination address of ICMP echo requests.	**destination ip** *ip-address*	By default, no destination IP address is specified.
5. Specify the payload size in each ICMP echo request.	**data-size** *size*	The default setting is 100 bytes.
6. Specify the string to be filled in the payload of each ICMP echo request.	**data-fill** *string*	The default setting is the hexadecimal number 00010203040506070809.
7. (Optional.) Specify the source IP address of ICMP echo requests.	• Specify the IP address of the specified interface as the source IP address: **source interface** *interface-type interface-number* • Specify the source IP address: **source ip** *ip-address*	By default, no source IP address is specified. The requests take the primary IP address of the output interface as their source IP address. If you configure both the **source ip** and **source interface** commands, the most recent configuration takes effect. The specified source interface must be up. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.
8. (Optional.) Specify the next hop for ICMP echo requests.	**next-hop** *ip-address*	By default, no next hop is configured.

Configuring the DHCP operation

The DHCP operation measures whether or not the DHCP server can respond to client requests. DHCP also measures the amount of time it takes the NQA client to obtain an IP address from a DHCP server.

The NQA client simulates the DHCP relay agent to forward DHCP requests for IP address acquisition from the DHCP server. The interface that performs the DHCP operation does not change its IP address. When the DHCP operation completes, the NQA client sends a packet to release the obtained IP address.

To configure the DHCP operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify the DHCP type and enter its view.	**type dhcp**	N/A
4. Specify the IP address of the DHCP server as the destination IP address of DHCP packets.	**destination ip** *ip-address*	By default, no destination IP address is specified.
5. (Optional.) Specify the source IP address of DHCP request packets.	**source ip** *ip-address*	By default, no source IP address is specified for the request packets. The requests take the IP address of the output interface as their source IP address. The specified source IP address must be the IP address of a local interface, and the local interface must be up. Otherwise, no probe packets can be sent out. The NQA client adds the source IP address to the **giaddr** field in DHCP requests to be sent to the DHCP server. For more information about the **giaddr** field, see *Layer 3—IP Services Configuration Guide*.

Configuring the DNS operation

The DNS operation measures the time for the NQA client to translate a domain name into an IP address through a DNS server.

A DNS operation simulates domain name resolution and does not save the obtained DNS entry.

To configure the DNS operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify the DNS type and enter its view.	**type dns**	N/A

Step		Command	Remarks
4.	Specify the IP address of the DNS server as the destination address of DNS packets.	**destination ip** *ip-address*	By default, no destination IP address is specified.
5.	Specify the domain name that needs to be translated.	**resolve-target** *domain-name*	By default, no domain name is specified.

Configuring the FTP operation

The FTP operation measures the time for the NQA client to transfer a file to or download a file from an FTP server.

When you configure the FTP operation, follow these restrictions and guidelines:

- When you perform the put operation with the **filename** command configured, make sure the file exists on the NQA client.
- If you get a file from the FTP server, make sure the file specified in the URL exists on the FTP server.
- The NQA client does not save the file obtained from the FTP server.
- Use a small file for the FTP operation. A big file might result in transfer failure because of timeout, or might affect other services for occupying much network bandwidth.

To configure the FTP operation:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3.	Specify the FTP type and enter its view.	**type ftp**	N/A
4.	Specify the URL of the destination FTP server.	**url** *url*	By default, no URL is specified for the destination FTP server. Enter the URL in one of the following formats: • ftp://*host*/*filename*. • ftp://*host*:*port*/*filename*. When you perform the **get** operation, the file name is required.
5.	(Optional.) Specify the source IP address of FTP request packets.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no FTP requests can be sent out.
6.	Specify the FTP operation type.	**operation** { **get** | **put** }	By default, the FTP operation type is **get**, which means obtaining files from the FTP server.

Step	Command	Remarks
7. Specify an FTP login username.	**username** *username*	By default, no FTP login username is configured.
8. Specify an FTP login password.	**password** { **cipher** | **simple** } *password*	By default, no FTP login password is configured.
9. (Optional.) Specify the name of a file to be transferred.	**filename** *file-name*	By default, no file is specified. This step is required if you perform the put operation.
10. Set the data transmission mode.	**mode** { **active** | **passive** }	The default mode is **active**.

Configuring the HTTP operation

An HTTP operation measures the time for the NQA client to obtain data from an HTTP server.

To configure an HTTP operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify the HTTP type and enter its view.	**type http**	N/A
4. Specify the URL of the destination HTTP server.	**url** *url*	By default, no URL is specified for the destination HTTP server. Enter the URL in one of the following formats: • http://*host/resource*. • http://*host:port/resource*.
5. Specify an HTTP login username.	**username** *username*	By default, no HTTP login username is specified.
6. Specify an HTTP login password.	**password** { **cipher** | **simple** } *password*	By default, no HTTP login password is specified.
7. (Optional.) Specify the source IP address of request packets.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no request packets can be sent out.
8. Specify the HTTP operation type.	**operation** { **get** | **post** | **raw** }	By default, the HTTP operation type is **get**, which means obtaining data from the HTTP server.
9. Specify the HTTP version.	**version** { **v1.0** | **v1.1** }	By default, HTTP 1.0 is used.

Step	Command	Remarks
10. (Optional.) Enter raw request view.	**raw-request**	Every time you enter raw request view, the previously configured content of the HTTP request is removed.
11. (Optional.) Specify the content of a GET request for the HTTP operation.	Enter or paste the content.	By default, no contents are specified. This step is required for the raw operation.
12. Save the input and exit to HTTP operation view.	**quit**	N/A

Configuring the UDP jitter operation

CAUTION:

To ensure successful UDP jitter operations and avoid affecting existing services, do not perform the operations on well-known ports from 1 to 1023.

Jitter means inter-packet delay variance. A UDP jitter operation measures unidirectional and bidirectional jitters. You can verify whether the network can carry jitter-sensitive services such as real-time voice and video services through the UDP jitter operation.

The UDP jitter operation works as follows:

1. The NQA client sends UDP packets to the destination port at a regular interval.
2. The destination device takes a time stamp to each packet that it receives, and then sends the packet back to the NQA client.
3. Upon receiving the responses, the NQA client calculates the jitter according to the time stamps.

The UDP jitter operation requires both the NQA server and the NQA client. Before you perform the UDP jitter operation, configure the UDP listening service on the NQA server. For more information about UDP listening service configuration, see "Configuring the NQA server."

To configure a UDP jitter operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify the UDP jitter type and enter its view.	**type udp-jitter**	N/A
4. Specify the destination address of UDP packets.	**destination ip** *ip-address*	By default, no destination IP address is specified. The destination IP address must be the same as the IP address of the listening service on the NQA server.

Step	Command	Remarks
5. Specify the destination port of UDP packets.	**destination port** *port-number*	By default, no destination port number is specified. The destination port number must be the same as the port number of the listening service on the NQA server.
6. (Optional.) Specify the source port number of UDP packets.	**source port** *port-number*	By default, no source port number is specified.
7. (Optional.) Specify the payload size in each UDP packet.	**data-size** *size*	The default setting is 100 bytes.
8. (Optional.) Specify the string to be filled in the payload of each UDP packet.	**data-fill** *string*	The default setting is the hexadecimal number 00010203040506070809.
9. (Optional.) Specify the number of UDP packets sent in one UDP jitter operation.	**probe packet-number** *packet-number*	The default setting is 10.
10. (Optional.) Configure the interval for sending UDP packets.	**probe packet-interval** *packet-interval*	The default setting is 20 milliseconds.
11. (Optional.) Specify how long the NQA client waits for a response from the server before it regards the response times out.	**probe packet-timeout** *packet-timeout*	The default setting is 3000 milliseconds.
12. (Optional.) Specify the source IP address for UDP packets.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no UDP packets can be sent out.

NOTE:

Use the **display nqa result** or **display nqa statistics** command to verify the UDP jitter operation. The **display nqa history** command does not display the UDP jitter operation results or statistics.

Configuring the SNMP operation

The SNMP operation measures the time for the NQA client to get a response packet from an SNMP agent.

To configure the SNMP operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step		Command	Remarks
2.	Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3.	Specify the SNMP type and enter its view.	**type snmp**	N/A
4.	Specify the destination address of SNMP packets.	**destination ip** *ip-address*	By default, no destination IP address is specified.
5.	(Optional.) Specify the source port of SNMP packets.	**source port** *port-number*	By default, no source port number is specified.
6.	(Optional.) Specify the source IP address of SNMP packets.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no SNMP packets can be sent out.

Configuring the TCP operation

The TCP operation measures the time for the NQA client to establish a TCP connection to a port on the NQA server.

The TCP operation requires both the NQA server and the NQA client. Before you perform a TCP operation, configure a TCP listening service on the NQA server. For more information about the TCP listening service configuration, see "Configuring the NQA server."

To configure the TCP operation:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3.	Specify the TCP type and enter its view.	**type tcp**	N/A
4.	Specify the destination address of TCP packets.	**destination ip** *ip-address*	By default, no destination IP address is specified. The destination address must be the same as the IP address of the listening service configured on the NQA server.
5.	Specify the destination port of TCP packets.	**destination port** *port-number*	By default, no destination port number is configured. The destination port number must be the same as the port number of the listening service on the NQA server.

Step	Command	Remarks
6. (Optional.) Specify the source IP address of TCP packets.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no TCP packets can be sent out.

Configuring the UDP echo operation

The UDP echo operation measures the round-trip time between the client and a UDP port on the NQA server.

The UDP echo operation requires both the NQA server and the NQA client. Before you perform a UDP echo operation, configure a UDP listening service on the NQA server. For more information about the UDP listening service configuration, see "Configuring the NQA server."

To configure the UDP echo operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify the UDP echo type and enter its view.	**type udp-echo**	N/A
4. Specify the destination address of UDP packets.	**destination ip** *ip-address*	By default, no destination IP address is specified. The destination address must be the same as the IP address of the listening service configured on the NQA server.
5. Specify the destination port of UDP packets.	**destination port** *port-number*	By default, no destination port number is specified. The destination port number must be the same as the port number of the listening service on the NQA server.
6. (Optional.) Specify the payload size in each UDP packet.	**data-size** *size*	The default setting is 100 bytes.
7. (Optional.) Specify the string to be filled in the payload of each UDP packet.	**data-fill** *string*	The default setting is the hexadecimal number 00010203040506070809.
8. (Optional.) Specify the source port of UDP packets.	**source port** *port-number*	By default, no source port number is specified.

Step	Command	Remarks
9. (Optional.) Specify the source IP address of UDP packets.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no UDP packets can be sent out.

Configuring the voice operation

△ CAUTION:

To ensure successful voice operations and avoid affecting existing services, do not perform the operations on well-known ports from 1 to 1023.

The voice operation measures VoIP network performance.

The voice operation works as follows:

1. The NQA client sends voice packets at sending intervals to the destination device (NQA server).

   The voice packets are of one of the following codec types:

   ○ G.711 A-law.

   ○ G.711 μ-law.

   ○ G.729 A-law.

2. The destination device takes a time stamp to each voice packet it receives and sends it back to the source.
3. Upon receiving the packet, the source device calculates the jitter and one-way delay based on the time stamp.

The following parameters that reflect VoIP network performance can be calculated by using the metrics gathered by the voice operation:

- **Calculated Planning Impairment Factor (ICPIF)**—Measures impairment to voice quality in a VoIP network. It is decided by packet loss and delay. A higher value represents a lower service quality.
- **Mean Opinion Scores (MOS)**—A MOS value can be evaluated by using the ICPIF value, in the range of 1 to 5. A higher value represents a higher service quality.

The evaluation of voice quality depends on users' tolerance for voice quality. For users with higher tolerance for voice quality, use the **advantage-factor** command to configure the advantage factor. When the system calculates the ICPIF value, it subtracts the advantage factor to modify ICPIF and MOS values for voice quality evaluation.

The voice operation requires both the NQA server and the NQA client. Before you perform a voice operation, configure a UDP listening service on the NQA server. For more information about UDP listening service configuration, see "Configuring the NQA server."

The voice operation cannot repeat.

To configure the voice operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step		Command	Remarks
2.	Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3.	Specify the voice type and enter its view.	**type voice**	N/A
4.	Specify the destination address of voice packets.	**destination ip** *ip-address*	By default, no destination IP address is configured. The destination IP address must be the same as the IP address of the listening service on the NQA server.
5.	Specify the destination port of voice packets.	**destination port** *port-number*	By default, no destination port number is configured. The destination port number must be the same as the port number of the listening service on the NQA server.
6.	(Optional.) Specify the codec type.	**codec-type** { **g711a** | **g711u** | **g729a** }	By default, the codec type is G.711 A-law.
7.	(Optional.) Specify the advantage factor for calculating MOS and ICPIF values.	**advantage-factor** *factor*	By default, the advantage factor is 0.
8.	(Optional.) Specify the source IP address of voice packets.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no voice packets can be sent out.
9.	(Optional.) Specify the source port number of voice packets.	**source port** *port-number*	By default, no source port number is specified.
10.	(Optional.) Specify the payload size in each voice packet.	**data-size** *size*	By default, the voice packet size varies by codec type. The default packet size is 172 bytes for G.711A-law and G.711 μ-law codec type, and 32 bytes for G.729 A-law codec type.
11.	(Optional.) Specify the string to be filled in the payload of each voice packet.	**data-fill** *string*	The default setting is the hexadecimal number 00010203040506070809.
12.	(Optional.) Specify the number of voice packets to be sent in a voice probe.	**probe packet-number** *packet-number*	The default setting is 1000.
13.	(Optional.) Specify the interval for sending voice packets.	**probe packet-interval** *packet-interval*	The default setting is 20 milliseconds.

Step	Command	Remarks
14. (Optional.) Specify how long the NQA client waits for a response from the server before it regards the response times out.	**probe packet-timeout** *packet-timeout*	The default setting is 5000 milliseconds.

NOTE:

Use the **display nqa result** or **display nqa statistics** command to verify the voice operation. The **display nqa history** command does not display the voice operation results or statistics.

Configuring the DLSw operation

The DLSw operation measures the response time of a DLSw device.

To configure the DLSw operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify the DLSw type and enter its view.	**type dlsw**	N/A
4. Specify the destination IP address of probe packets.	**destination ip** *ip-address*	By default, no destination IP address is specified.
5. (Optional.) Specify the source IP address of probe packets.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

Configuring the path jitter operation

The path jitter operation measures the jitter, negative jitters, and positive jitters from the NQA client to each hop on the path to the destination.

Before you configure the path jitter operation, perform the following tasks:

- Enable sending ICMP time exceeded messages on the intermediate devices between the source and the destination devices. If the intermediate devices are HP devices, use the **ip ttl-expires enable** command.
- Enable sending ICMP destination unreachable messages on the destination device. If the destination device is an HP device, use the **ip unreachables enable** command.

For more information about the **ip ttl-expires enable** and **ip unreachable enable** commands, see *Layer 3—IP Services Command Reference*.

To configure the path jitter operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify the path jitter type and enter its view.	**type path-jitter**	N/A
4. Specify the destination address of ICMP echo requests.	**destination ip** *ip-address*	By default, no destination IP address is specified.
5. (Optional.) Specify the payload size in each ICMP echo request.	**data-size** *size*	The default setting is 100 bytes.
6. (Optional.) Specify the string to be filled in the payload of each ICMP echo request.	**data-fill** *string*	The default setting is the hexadecimal number 00010203040506070809.
7. (Optional.) Specify the source IP address of ICMP echo requests.	**source ip** *ip-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no ICMP echo requests can be sent out.
8. (Optional.) Specify the number of ICMP echo requests to be sent in a path jitter operation.	**probe packet-number** *packet-number*	The default setting is 10.
9. (Optional.) Specify the interval for sending ICMP echo requests.	**probe packet-interval** *packet-interval*	The default setting is 20 milliseconds.
10. (Optional.) Specify how long the NQA client waits for a response from the server before it regards the response times out.	**probe packet-timeout** *packet-timeout*	The default setting is 3000 milliseconds.
11. (Optional.) Specify an LSR path.	**lsr-path** *ip-address*&<1-8>	By default, no LSR path is specified. The path jitter operation uses the tracert to detect the LSR path to the destination, and sends ICMP echo requests to each hop on the LSR.
12. (Optional.) Perform the path jitter operation only on the destination address.	**target-only**	By default, the path jitter operation is performed on each hop on the path to the destination.

Configuring optional parameters for the NQA operation

Unless otherwise specified, the following optional parameters apply to all types of NQA operations.

To configure optional parameters for an NQA operation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify an NQA operation type and enter its view.	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **path-jitter** | **snmp** | **tcp** | **udp-echo** | **udp-jitter** | **voice** }	N/A
4. Configure a description.	**description** *text*	By default, no description is configured.
5. Specify the interval at which the NQA operation repeats.	**frequency** *interval*	For a voice or path jitter operation, the default setting is 60000 milliseconds. For other operations, the default setting is 0 milliseconds. Only one operation is performed. If the operation is not completed when the interval expires, the next operation does not start.
6. Specify the probe times.	**probe count** *times*	By default, an NQA operation performs one probe. This command is not available for the path jitter and voice operations. Each of these operations performs only one probe.
7. Specify the probe timeout time.	**probe timeout** *timeout*	The default setting is 3000 milliseconds. This command is not available for the path jitter, UDP jitter, and voice operations.
8. Specify the TTL for probe packets.	**ttl** *value*	The default setting is 20. This command is not available for the DHCP and path jitter operations.
9. Specify the ToS value in the IP header for probe packets.	**tos** *value*	The default setting is 0.
10. Enable the routing table bypass function.	**route-option bypass-route**	By default, the routing table bypass function is disabled. This command is not available for the DHCP and path jitter operations.

Configuring the collaboration function

Collaboration is implemented by associating a reaction entry of an NQA operation with a track entry. The reaction entry monitors the NQA operation. If the number of operation failures reaches the specified threshold, the configured action is triggered.

To configure the collaboration function:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify an NQA operation type and enter its view.	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **snmp** | **tcp** | **udp-echo** }	The collaboration function is not available for the path jitter, UDP jitter, and voice operations.
4. Configure a reaction entry.	**reaction** *item-number* **checked-element probe-fail threshold-type consecutive** *consecutive-occurrences* **action-type trigger-only**	By default, no reaction entry is configured. You cannot modify the content of an existing reaction entry.
5. Exit to system view.	**quit**	N/A
6. Associate Track with NQA.	See *High Availability Configuration Guide*.	N/A
7. Associate Track with an application module.	See *High Availability Configuration Guide*.	N/A

Configuring threshold monitoring

This function allows you to monitor the NQA operation running status.

Threshold types

An NQA operation supports the following threshold types:

- **average**—If the average value for the monitored performance metric either exceeds the upper threshold or goes below the lower threshold, a threshold violation occurs.
- **accumulate**—If the total number of times that the monitored performance metric is out of the specified value range reaches or exceeds the specified threshold, a threshold violation occurs.
- **consecutive**—If the number of consecutive times that the monitored performance metric is out of the specified value range reaches or exceeds the specified threshold, a threshold violation occurs.

Threshold violations for the average or accumulate threshold type are determined on a per NQA operation basis. The threshold violations for the consecutive type are determined from the time the NQA operation starts.

Triggered actions

The following actions might be triggered:

- **none**—NQA displays results only on the terminal screen. It does not send traps to the NMS.
- **trap-only**—NQA displays results on the terminal screen, and meanwhile it sends traps to the NMS.
- **trigger-only**—NQA displays results on the terminal screen, and meanwhile triggers other modules for collaboration.

The DNS operation does not support the action of sending trap messages.

Reaction entry

In a reaction entry, configure a monitored element, a threshold type, and an action to be triggered to implement threshold monitoring.

The state of a reaction entry can be invalid, over-threshold, or below-threshold.

- Before an NQA operation starts, the reaction entry is in invalid state.
- If the threshold is violated, the state of the entry is set to over-threshold. Otherwise, the state of the entry is set to below-threshold.

If the action is configured as **trap-only** for a reaction entry, a trap message is sent to the NMS when the state of the entry changes.

Configuration procedure

Before you configure threshold monitoring, configure the destination address of the trap messages by using the **snmp-agent target-host** command. For more information about the command, see *Network Management and Monitoring Command Reference*.

To configure threshold monitoring:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3.	Enter NQA operation view.	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **snmp** | **tcp** | **udp-echo** | **udp-jitter** | **voice** }	Path jitter does not support threshold monitoring.
4.	Enable sending traps to the NMS when specific conditions are met.	**reaction trap** { **probe-failure** *consecutive-probe-failures* | **test-complete** | **test-failure** *cumulate-probe-failures* }	By default, no traps are sent to the NMS. The UDP jitter and voice operations support only the **test-complete** keyword.

Step	Command	Remarks
5. Configure threshold monitoring.	• Monitor the operation duration (not supported in the UDP jitter and voice operations): **reaction** *item-number* **checked-element probe-duration threshold-type** { **accumulate** *accumulate-occurrences* | **average** | **consecutive** *consecutive-occurrences* } **threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }] • Monitor failure times (not supported in the UDP jitter and voice operations): **reaction** *item-number* **checked-element probe-fail threshold-type** { **accumulate** *accumulate-occurrences* | **consecutive** *consecutive-occurrences* } [**action-type** { **none** | **trap-only** }] • Monitor the round-trip time (only for the in UDP jitter and voice operations): **reaction** *item-number* **checked-element rtt threshold-type** { **accumulate** *accumulate-occurrences* | **average** } **threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }] • Monitor packet loss (only for the UDP jitter and voice operations): **reaction** *item-number* **checked-element packet-loss threshold-type accumulate** *accumulate-occurrences* [**action-type** { **none** | **trap-only** }] • Monitor the one-way jitter (only for the UDP jitter and voice operations): **reaction** *item-number* **checked-element** { **jitter-ds** | **jitter-sd** } **threshold-type** { **accumulate** *accumulate-occurrences* | **average** } **threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }] • Monitor the one-way delay (only for the UDP jitter and voice operations): **reaction** *item-number* **checked-element** { **owd-ds** | **owd-sd** } **threshold-value** *upper-threshold lower-threshold* • Monitor the ICPIF value (only for the voice operation): **reaction** *item-number* **checked-element icpif threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }] • Monitor the MOS value (only for the voice operation): **reaction** *item-number* **checked-element mos threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }]	N/A

Configuring the NQA statistics collection function

NQA forms statistics within the same collection interval as a statistics group. To display information about the statistics groups, use the **display nqa statistics** command.

NQA does not generate any statistics group for the operation that runs once. To set the NQA operation to run only once, use the **frequency** command to set the interval to 0 milliseconds.

To configure the NQA statistics collection function:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Specify an NQA operation type and enter its view.	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **path-jitter** | **snmp** | **tcp** | **udp-echo** | **udp-jitter** | **voice** }	DHCP operation does not support the NQA statistics collection function.
4. (Optional.) Specify the interval for collecting the statistics.	**statistics interval** *interval*	The default setting is 60 minutes.
5. (Optional.) Specify the maximum number of statistics groups that can be saved.	**statistics max-group** *number*	The default setting is two groups. To disable collecting NQA statistics, set the maximum number to 0. When the maximum number of statistics groups is reached, to save a new statistics group, the oldest statistics group is deleted.
6. (Optional.) Specify the hold time of statistics groups.	**statistics hold-time** *hold-time*	The default setting is 120 minutes. A statistics group is deleted when its hold time expires.

Configuring the saving of NQA history records

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an NQA operation and enter NQA operation view.	**nqa entry** *admin-name operation-tag*	By default, no NQA operation is created.
3. Enter NQA operation type view.	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **snmp** | **tcp** | **udp-echo** }	The UDP jitter, path jitter, and voice operations do not support saving history records.

Step	Command	Remarks
4. Enable the saving of history records for the NQA operation.	**history-record enable**	By default, this function is not enabled.
5. (Optional.) Set the lifetime of history records.	**history-record keep-time** *keep-time*	The default setting is 120 minutes. A record is deleted when its lifetime is reached.
6. (Optional.) Specify the maximum number of history records that can be saved.	**history-record number** *number*	The default setting is 50. If the maximum number of history records for an NQA operation is reached, the earliest history records are deleted.
7. (Optional.) Display NQA history records.	**display nqa history**	N/A

Scheduling the NQA operation on the NQA client

The NQA operation works between the specified start time and the end time (the start time plus operation duration). If the specified start time is ahead of the system time, the operation starts immediately. If both the specified start and end time are ahead of the system time, the operation does not start. To display the current system time, use the **display clock** command.

When you schedule an NQA operation, follow these restrictions and guidelines:

- You cannot enter the operation type view or the operation view of a scheduled NQA operation.
- A system time adjustment does not affect started or completed NQA operations. It affects only the NQA operations that have not started.

To schedule the NQA operation on the NQA client:

Step	Command
1. Enter system view.	**system-view**
2. Specify the scheduling parameters for an NQA operation.	**nqa schedule** *admin-name operation-tag* **start-time** { *hh:mm:ss* [*yyyy/mm/dd* | *mm/dd/yyyy*] | **now** } **lifetime** { *lifetime* | **forever** } [**recurring**]

Configuring NQA templates on the NQA client

An NQA template is a set of operation parameters, such as the destination address, the destination port number, and the destination server URL. You can use an NQA template in a feature to provide statistics. You can create multiple templates on a device, and each template must be uniquely named.

NQA template supports the ICMP, DNS, HTTP, TCP, and FTP operation types.

Configuring the ICMP template

A feature that uses the ICMP template performs the ICMP operation to measure the reachability of a destination device. The ICMP template is supported in both IPv4 and IPv6 networks.

To configure the ICMP template:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an ICMP template and enter its view.	**nqa template icmp** *name*	N/A
3. (Optional.) Specify the destination IPv4 or IPv6 address of the operation.	• IPv4 address: **destination ip** *ip-address* • IPv6 address: **destination ipv6** *ipv6-address*	By default, no destination IP address is configured.
4. Specify the payload size in each ICMP request.	**data-size** *size*	The default setting is 100 bytes.
5. Specify the string to be filled in the payload of each request.	**data-fill** *string*	The default setting is the hexadecimal number 00010203040506070809.
6. (Optional.) Specify the IP address of the specified interface as the source IP address of ICMP echo requests.	**source interface** *interface-type interface-number*	By default, no source IP address is specified. The requests use the primary IP address of the output interface as their source IP address. The specified source interface must be up. If you configure the **source interface** command with the **source ip** or **source ipv6** command, the most recent configuration takes effect.
7. (Optional.) Specify the source IPv4 or IPv6 address for the probe packets.	• IPv4 address: **source ip** *ip-address* • IPv6 address: **source ipv6** *ipv6-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

Configuring the DNS template

A feature that uses the DNS template performs the DNS operation to determine the status of the server. It is supported in both IPv4 and IPv6 networks.

In DNS template view, you can specify the address expected to be returned. If the returned IP addresses include the expected address, the DNS server is valid and the operation succeeds. Otherwise, the operation fails.

Create a mapping between the domain name and an address before you perform the DNS operation. For information about configuring the DNS server, see *Layer 3—IP Services Configuration Guide*.

To configure the DNS template:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Create a DNS template and enter DNS template view.	**nqa template dns** *name*	N/A
3.	(Optional.) Specify the destination IPv4 or IPv6 address of DNS packets.	• IPv4 address: **destination ip** *ip-address* • IPv6 address: **destination ipv6** *ipv6-address*	By default, no destination address is specified.
4.	Configure the destination port number for the operation.	**destination port** *port-number*	By default, the destination port number is 53.
5.	Specify the domain name that needs to be translated.	**resolve-target** *domain-name*	By default, no domain name is specified.
6.	Configure the domain name resolution type.	**resolve-type** { **A** | **AAAA** }	By default, the type is type A. A type A query resolves a domain name to a mapped IPv4 address, and a type AAAA query to a mapped IPv6 address.
7.	(Optional.) Specify the source IPv4 or IPv6 address for the probe packets.	• IPv4 address: **source ip** *ip-address* • IPv6 address: **source ipv6** *ipv6-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.
8.	(Optional.) Configure the source port for probe packets.	**source port** *port-number*	By default, no source port number is configured.
9.	(Optional.) Specify the IPv4 or IPv6 address that is expected to be returned.	• IPv4 address: **expect ip** *ip-address* • IPv6 address: **expect ipv6** *ipv6-address*	By default, no expected IP address is specified.

Configuring the TCP template

A feature that uses the TCP template performs the TCP operation to test the following items:

- Whether the NQA client can establish a TCP connection to a specific port on the server.
- Whether the requested service is available on the server.

In TCP template view, you can specify the expected data to be returned. If you do not specify the expected data, the TCP operation tests only whether the client can establish a TCP connection to the server.

The TCP operation requires both the NQA server and the NQA client. Before you perform a TCP operation, configure a TCP listening service on the NQA server. For more information about the TCP listening service configuration, see "Configuring the NQA server."

To configure the TCP template:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a TCP template and enter its view.	**nqa template tcp** *name*	N/A
3. (Optional.) Specify the destination IPv4 or IPv6 address of the operation.	• IPv4 address: **destination ip** *ip-address* • IPv6 address: **destination ipv6** *ipv6-address*	By default, no destination address is specified. The destination address must be the same as the IP address of the listening service configured on the NQA server.
4. (Optional.) Configure the destination port number for the operation.	**destination port** *port-number*	By default, no destination port number is configured. The destination port number must be the same as the port number of the listening service on the NQA server.
5. Specify the string to be filled in the payload of each request.	**data-fill** *string*	The default setting is the hexadecimal number 00010203040506070809.
6. (Optional.) Specify the source IPv4 or IPv6 address for the probe packets.	• IPv4 address: **source ip** *ip-address* • IPv6 address: **source ipv6** *ipv6-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.
7. (Optional.) Configure the expected data.	**expect data** *expression* [**offset** *number*]	By default, no expected data is configured. The expected data is checked only when you configure both the **data-fill** and **expect-data** commands.

Configuring the HTTP template

A feature that uses the HTTP template performs the HTTP operation to measure the time it takes the NQA client to obtain data from an HTTP server.

The expected data is checked only when the expected data is configured and the HTTP response contains the Content-Length field in the HTTP header. The Content-Length field indicates the packet body length, and it does not include the header length. An HTTP packet with this field indicates that the packet data does not include the multipart type and the packet body is a data type.

The status code of the HTTP packet is a three-digit field in decimal notation, and it includes the status information for the HTTP server. The first digit defines the class of response, and the last two digits do not have any categorization role.

Configure the HTTP server before you perform the HTTP operation.

To configure the HTTP template:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an HTTP template and enter its view.	**nqa template http** *name*	N/A
3. Specify the URL of the destination HTTP server.	**url** *url*	By default, no URL is specified for the destination HTTP server. Enter the URL in one of the following formats: - http://*host*/*resource*. - http://*host*:*port*/*resource*.
4. Specify an HTTP login username.	**username** *username*	By default, no HTTP login username is specified.
5. Specify an HTTP login password.	**password** { **cipher** | **simple** } *password*	By default, no HTTP login password is specified.
6. Specify the HTTP operation type.	**operation** { **get** | **post** | **raw** }	By default, the HTTP operation type is **get**, which means obtaining data from the HTTP server. In the HTTP raw operation, use the **raw-request** command to specify the content of the GET request to be sent to the HTTP server.
7. (Optional.) Enter raw request view.	**raw-request**	This step is required for the raw operation. Every time you enter the raw request view, the previously configured content of the GET request is removed.
8. (Optional.) Enter or paste the content of the GET request for the HTTP operation.	N/A	This step is required for the raw operation. By default, no contents are specified.
9. (Optional.) Save the input and exit to HTTP template view.	**quit**	N/A
10. (Optional.) Specify the source IPv4 or IPv6 address for the probe packets.	- IPv4 address: **source ip** *ip-address* - IPv6 address: **source ipv6** *ipv6-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.
11. (Optional.) Configure the expected status codes.	**expect status** *status-list*	By default, no expected status code is configured.
12. (Optional.) Configure the expected data.	**expect data** *expression* [**offset** *number*]	By default, no expected data is configured.

Configuring the FTP template

A feature that uses the FTP template performs the FTP operation. The operation measures the time it takes the NQA client to transfer a file to or download a file from an FTP server.

Configure the username and password for the FTP client to log in to the FTP server before you perform an FTP operation. For information about configuring the FTP server, see *Fundamentals Configuration Guide*.

To configure the FTP template:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an FTP template and enter its view.	**nqa template ftp** *name*	N/A
3. Specify the URL of the destination FTP server.	**url** *url*	By default, no URL is specified for the destination FTP server. Enter the URL in one of the following formats: • ftp://*host*/*filename*. • ftp://*host*:*port*/*filename*. When you perform the **get** operation, the file name is required. When you perform the **put** operation, the *filename* argument does not take effect, even if it is specified. The file name for the **put** operation is determined by the **filename** command.
4. (Optional.) Specify the FTP operation type.	**operation** { **get** | **put** }	By default, the FTP operation type is **get**, which means obtaining files from the FTP server.
5. Specify an FTP login username.	**username** *username*	By default, no FTP login username is specified.
6. Specify an FTP login password.	**password** { **cipher** | **simple** } *password*	By default, no FTP login password is specified.
7. (Optional.) Specify the name of a file to be transferred.	**filename** *filename*	By default, no file is specified. This step is required if you perform the **put** operation. This configuration does not take effect for the **get** operation.
8. Set the data transmission mode.	**mode** { **active** | **passive** }	The default mode is **active**.
9. (Optional.) Specify the source IPv4 or IPv6 address for the probe packets.	• IPv4 address: **source ip** *ip-address* • IPv6 address: **source ipv6** *ipv6-address*	By default, no source IP address is specified. The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

Configuring optional parameters for the NQA template

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Create an NQA template and enter its view.	**nqa template** { **dns** | **ftp** | **http** | **icmp** | **tcp** } *name*	N/A
3.	Configure a description.	**description** *text*	By default, no description is configured.
4.	Specify the interval at which the NQA operation repeats.	**frequency** *interval*	The default setting is 5000 milliseconds. If the operation is not completed when the interval expires, the next operation does not start.
5.	Specify the probe timeout time.	**probe timeout** *timeout*	The default setting is 3000 milliseconds.
6.	Specify the TTL for probe packets.	**ttl** *value*	The default setting is 20.
7.	Specify the ToS value in the IP header for probe packets.	**tos** *value*	The default setting is 0.
8.	Configure the number of consecutive successful probes that lead to a successful operation.	**reaction trigger probe-pass** *count*	The default setting is 3. If the number of consecutive successful probes for an NQA operation is reached, the NQA client notifies the feature that uses the template of the successful operation event.
9.	Configure the number of consecutive probe failures that lead to an operation failure.	**reaction trigger probe-fail** *count*	The default setting is 3. If the number of consecutive probe failures for an NQA operation is reached, the NQA client notifies the feature that uses the NQA template of the operation failure.

Displaying and maintaining NQA

Execute **display** commands in any view.

Task	Command
Display history records of NQA operations.	**display nqa history** [*admin-name operation-tag*]
Display the current monitoring results of reaction entries.	**display nqa reaction counters** [*admin-name operation-tag* [*item-number*]]
Display the most recent result of the NQA operation.	**display nqa result** [*admin-name operation-tag*]
Display NQA statistics.	**display nqa statistics** [*admin-name operation-tag*]
Display NQA server status.	**display nqa server status**

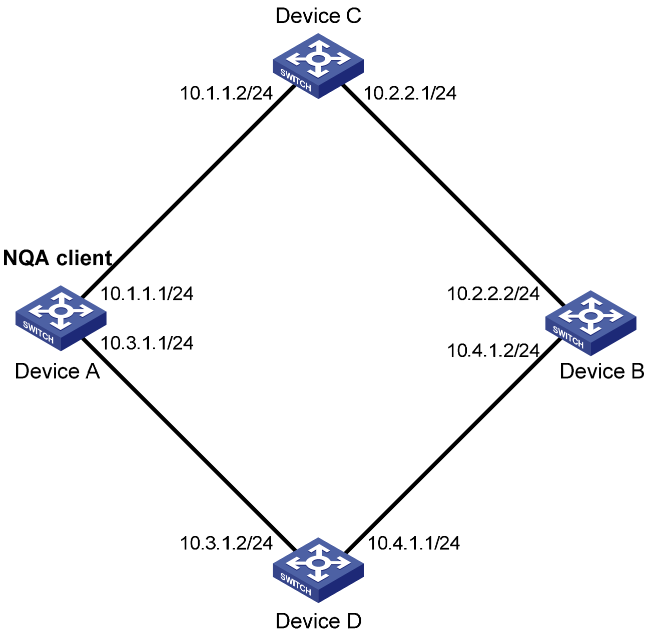
NQA configuration examples

ICMP echo operation configuration example

Network requirements

As shown in Figure 38, configure an ICMP echo operation from the NQA client Device A to Device B to test the round-trip time. The next hop of Device A is Device C.

Figure 38 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create an ICMP echo operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type icmp-echo
```

# Specify the destination IP address of ICMP echo requests as 10.2.2.2.

```
[DeviceA-nqa-admin-test1-icmp-echo] destination ip 10.2.2.2
```

# Configure 10.1.1.2 as the next hop. The ICMP echo requests are sent through Device C to Device B.

```
[DeviceA-nqa-admin-test1-icmp-echo] next-hop 10.1.1.2
```

# Configure the ICMP echo operation to perform 10 probes.

```
[DeviceA-nqa-admin-test1-icmp-echo] probe count 10
```

# Specify the probe timeout time for the ICMP echo operation as 500 milliseconds.

```
[DeviceA-nqa-admin-test1-icmp-echo] probe timeout 500
```

# Configure the ICMP echo operation to repeat at an interval of 5000 milliseconds.

```
[DeviceA-nqa-admin-test1-icmp-echo] frequency 5000
```

# Enable saving history records.

```
[DeviceA-nqa-admin-test1-icmp-echo] history-record enable
```

# Configure the maximum number of history records that can be saved as 10.

```
[DeviceA-nqa-admin-test1-icmp-echo] history-record number 10
[DeviceA-nqa-admin-test1-icmp-echo] quit
```

# Start the ICMP echo operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

# After the ICMP echo operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

# Display the most recent result of the ICMP echo operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 10             Receive response times: 10
    Min/Max/Average round trip time: 2/5/3
    Square-Sum of round trip time: 96
    Last succeeded probe time: 2011-08-23 15:00:01.2
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

# Display the history records of the ICMP echo operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test) history records:
  Index      Response     Status           Time
  370        3            Succeeded        2007-08-23 15:00:01.2
  369        3            Succeeded        2007-08-23 15:00:01.2
  368        3            Succeeded        2007-08-23 15:00:01.2
  367        5            Succeeded        2007-08-23 15:00:01.2
  366        3            Succeeded        2007-08-23 15:00:01.2
  365        3            Succeeded        2007-08-23 15:00:01.2
  364        3            Succeeded        2007-08-23 15:00:01.1
  363        2            Succeeded        2007-08-23 15:00:01.1
  362        3            Succeeded        2007-08-23 15:00:01.1
  361        2            Succeeded        2007-08-23 15:00:01.1
```

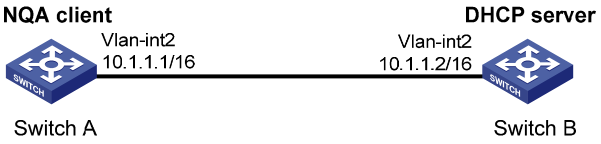
The output shows that the packets sent by Device A can reach Device B through Device C. No packet loss occurs during the operation. The minimum, maximum, and average round-trip times are 2, 5, and 3 milliseconds, respectively.

DHCP operation configuration example

Network requirements

As shown in Figure 39, configure a DHCP operation to test the time required for Switch A to obtain an IP address from the DHCP server.

Figure 39 Network diagram



Configuration procedure

# Create a DHCP operation.

```
<SwitchA> system-view
[SwitchA] nqa entry admin test1
[SwitchA-nqa-admin-test1] type dhcp
```

# Specify the DHCP server IP address 10.1.1.2 as the destination address.

```
[SwitchA-nqa-admin-test1-dhcp] destination ip 10.1.1.2
```

# Enable the saving of history records.

```
[SwitchA-nqa-admin-test1-dhcp] history-record enable
[SwitchA-nqa-admin-test1-dhcp] quit
```

# Start the DHCP operation.

```
[SwitchA] nqa schedule admin test1 start-time now lifetime forever
```

# After the DHCP operation runs for a period of time, stop the operation.

```
[SwitchA] undo nqa schedule admin test1
```

# Display the most recent result of the DHCP operation.

```
[SwitchA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 512/512/512
    Square-Sum of round trip time: 262144
    Last succeeded probe time: 2011-11-22 09:56:03.2
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

# Display the history records of the DHCP operation.

```
[SwitchA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          512          Succeeded        2011-11-22 09:56:03.2
```

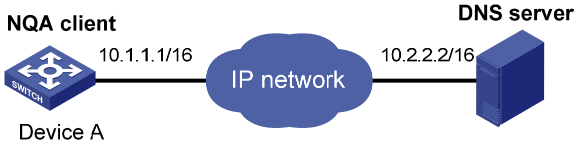
The output shows that it took Switch A 512 milliseconds to obtain an IP address from the DHCP server.

DNS operation configuration example

Network requirements

As shown in Figure 40, configure a DNS operation to test whether Device A can perform address resolution through the DNS server and test the resolution time.

Figure 40 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create a DNS operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type dns
```

# Specify the IP address of the DNS server 10.2.2.2 as the destination address.

```
[DeviceA-nqa-admin-test1-dns] destination ip 10.2.2.2
```

# Specify the domain name to be translated as **host.com**.

```
[DeviceA-nqa-admin-test1-dns] resolve-target host.com
```

# Enable the saving of history records.

```
[DeviceA-nqa-admin-test1-dns] history-record enable
[DeviceA-nqa-admin-test1-dns] quit
```

# Start the DNS operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

# After the DNS operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

# Display the most recent result of the DNS operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 62/62/62
    Square-Sum of round trip time: 3844
    Last succeeded probe time: 2011-11-10 10:49:37.3
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

# Display the history records of the DNS operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test) history records:
  Index      Response     Status           Time
  1          62           Succeeded        2011-11-10 10:49:37.3
```

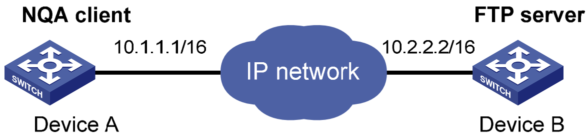
The output shows that it took Device A 62 milliseconds to translate domain name **host.com** into an IP address.

FTP operation configuration example

Network requirements

As shown in Figure 41, configure an FTP operation to test the time required for Device A to upload a file to the FTP server. The login username and password are **admin** and **systemtest**, respectively. The file to be transferred to the FTP server is **config.txt**.

Figure 41 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create an FTP operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type ftp
```

# Specify the URL of the FTP server.

```
[DeviceA-nqa-admin-test-ftp] url ftp://10.2.2.2
```

# Specify 10.1.1.1 as the source IP address.

```
[DeviceA-nqa-admin-test1-ftp] source ip 10.1.1.1
```

# Configure the device to upload file **config.txt** to the FTP server.

```
[DeviceA-nqa-admin-test1-ftp] operation put
[DeviceA-nqa-admin-test1-ftp] filename config.txt
```

# Specify the username for the FTP operation as **admin**.

```
[DeviceA-nqa-admin-test1-ftp] username admin
```

# Specify the password for the FTP operation as **systemtest**.

```
[DeviceA-nqa-admin-test1-ftp] password simple systemtest
```

# Enable the saving of history records.

```
[DeviceA-nqa-admin-test1-ftp] history-record enable
[DeviceA-nqa-admin-test1-ftp] quit
```

# Start the FTP operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

# After the FTP operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

# Display the most recent result of the FTP operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 173/173/173
    Square-Sum of round trip time: 29929
    Last succeeded probe time: 2011-11-22 10:07:28.6
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to disconnect: 0
    Failures due to no connection: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

# Display the history records of the FTP operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          173          Succeeded        2011-11-22 10:07:28.6
```

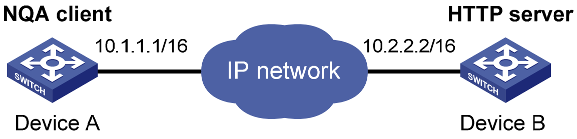
The output shows that it took Device A 173 milliseconds to upload a file to the FTP server.

HTTP operation configuration example

Network requirements

As shown in Figure 42, configure an HTTP operation on the NQA client to test the time required to obtain data from the HTTP server.

Figure 42 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create an HTTP operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type http
```

# Specify the URL of the HTTP server.

```
[DeviceA-nqa-admin-test-http] url http://10.2.2.2/index.htm
```

# Configure the HTTP operation to get data from the HTTP server.

```
[DeviceA-nqa-admin-test1-http] operation get
```

# Configure the operation to use HTTP version 1.0.

```
[DeviceA-nqa-admin-test1-http] version v1.0
```

# Enable the saving of history records.

```
[DeviceA-nqa-admin-test1-http] history-record enable
[DeviceA-nqa-admin-test1-http] quit
```

# Start the HTTP operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

# After the HTTP operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

# Display the most recent result of the HTTP operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 64/64/64
    Square-Sum of round trip time: 4096
    Last succeeded probe time: 2011-11-22 10:12:47.9
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to disconnect: 0
    Failures due to no connection: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

# Display the history records of the HTTP operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          64           Succeeded        2011-11-22 10:12:47.9
```

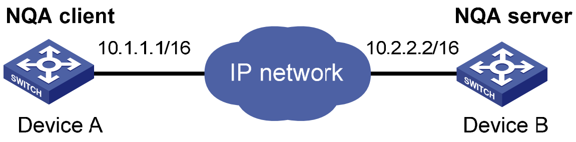
The output shows that it took Device A 64 milliseconds to obtain data from the HTTP server.

UDP jitter operation configuration example

Network requirements

As shown in Figure 43, configure a UDP jitter operation to test the jitter, delay, and round-trip time between Device A and Device B.

Figure 43 Network diagram



Configuration procedure

1. Assign each interface an IP address. (Details not shown.)
2. Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)
3. Configure Device B:

   # Enable the NQA server.

   ```
   <DeviceB> system-view
   [DeviceB] nqa server enable
   ```

   # Configure a listening service to listen on the IP address 10.2.2.2 and UDP port 9000.

   ```
   [DeviceB] nqa server udp-echo 10.2.2.2 9000
   ```

4. Configure Device A:

   # Create a UDP jitter operation.

   ```
   <DeviceA> system-view
   [DeviceA] nqa entry admin test1
   [DeviceA-nqa-admin-test1] type udp-jitter
   ```

   # Configure 10.2.2.2 as the destination IP address and port 9000 as the destination port.

   ```
   [DeviceA-nqa-admin-test1-udp-jitter] destination ip 10.2.2.2
   [DeviceA-nqa-admin-test1-udp-jitter] destination port 9000
   ```

   # Configure the operation to repeat at an interval of 1000 milliseconds.

   ```
   [DeviceA-nqa-admin-test1-udp-jitter] frequency 1000
   [DeviceA-nqa-admin-test1-udp-jitter] quit
   ```

   # Start the UDP jitter operation.

   ```
   [DeviceA] nqa schedule admin test1 start-time now lifetime forever
   ```

   # After the UDP jitter operation runs for a period of time, stop the operation.

   ```
   [DeviceA] undo nqa schedule admin test1
   ```

   # Display the most recent result of the UDP jitter operation.

   ```
   [DeviceA] display nqa result admin test1
   NQA entry (admin admin, tag test1) test results:
       Send operation times: 10             Receive response times: 10
       Min/Max/Average round trip time: 15/32/17
       Square-Sum of round trip time: 3235
       Last packet received time: 2011-05-29 13:56:17.6
     Extended results:
       Packet loss ratio: 0%
       Failures due to timeout: 0
       Failures due to internal error: 0
       Failures due to other errors: 0
       Packets out of sequence: 0
       Packets arrived late: 0
     UDP-jitter results:
      RTT number: 10
       Min positive SD: 4                     Min positive DS: 1
       Max positive SD: 21                    Max positive DS: 28
       Positive SD number: 5                  Positive DS number: 4
       Positive SD sum: 52                    Positive DS sum: 38
   ```

```
    Positive SD average: 10                Positive DS average: 10
    Positive SD square-sum: 754            Positive DS square-sum: 460
    Min negative SD: 1                     Min negative DS: 6
    Max negative SD: 13                    Max negative DS: 22
    Negative SD number: 4                  Negative DS number: 5
    Negative SD sum: 38                    Negative DS sum: 52
    Negative SD average: 10                Negative DS average: 10
    Negative SD square-sum: 460            Negative DS square-sum: 754
  One way results:
    Max SD delay: 15                       Max DS delay: 16
    Min SD delay: 7                        Min DS delay: 7
    Number of SD delay: 10                 Number of DS delay: 10
    Sum of SD delay: 78                    Sum of DS delay: 85
    Square-Sum of SD delay: 666            Square-Sum of DS delay: 787
    SD lost packets: 0                   DS lost packets: 0
    Lost packets for unknown reason: 0
```

# Display the statistics of the UDP jitter operation.

```
[DeviceA] display nqa statistics admin test1
NQA entry (admin admin, tag test1) test statistics:
  NO. : 1
    Start time: 2011-05-29 13:56:14.0
    Life time: 47 seconds
    Send operation times: 410            Receive response times: 410
    Min/Max/Average round trip time: 1/93/19
    Square-Sum of round trip time: 206176
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
    Packets out of sequence: 0
    Packets arrived late: 0
  UDP-jitter results:
   RTT number: 410
    Min positive SD: 3                     Min positive DS: 1
    Max positive SD: 30                    Max positive DS: 79
    Positive SD number: 186                Positive DS number: 158
    Positive SD sum: 2602                  Positive DS sum: 1928
    Positive SD average: 13                Positive DS average: 12
    Positive SD square-sum: 45304          Positive DS square-sum: 31682
    Min negative SD: 1                     Min negative DS: 1
    Max negative SD: 30                    Max negative DS: 78
    Negative SD number: 181                Negative DS number: 209
    Negative SD sum: 181                   Negative DS sum: 209
    Negative SD average: 13                Negative DS average: 14
    Negative SD square-sum: 46994          Negative DS square-sum: 3030
  One way results:
    Max SD delay: 46                       Max DS delay: 46
    Min SD delay: 7                        Min DS delay: 7
    Number of SD delay: 410                Number of DS delay: 410
    Sum of SD delay: 3705                  Sum of DS delay: 3891
    Square-Sum of SD delay: 45987          Square-Sum of DS delay: 49393
    SD lost packets: 0                   DS lost packets: 0
    Lost packets for unknown reason: 0
```

SNMP operation configuration example

Network requirements

As shown in Figure 44, configure an SNMP operation to test the time the NQA client uses to get a response from the SNMP agent.

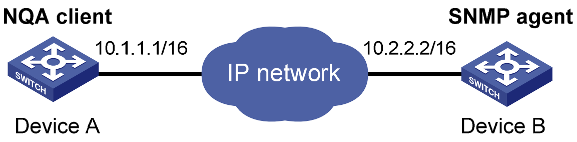


Figure 44 Network diagram

Configuration procedure

1. Assign each interface an IP address. (Details not shown.)
2. Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)
3. Configure the SNMP agent (Device B):

   # Set the SNMP version to **all**.

   ```
   <DeviceB> system-view
   [DeviceB] snmp-agent sys-info version all
   ```

   # Set the read community to **public**.

   ```
   [DeviceB] snmp-agent community read public
   ```

   # Set the write community to **private**.

   ```
   [DeviceB] snmp-agent community write private
   ```

4. Configure Device A:

   # Create an SNMP operation.

   ```
   <DeviceA> system-view
   [DeviceA] nqa entry admin test1
   [DeviceA-nqa-admin-test1] type snmp
   ```

   # Configure 10.2.2.2 as the destination IP address of the SNMP operation.

   ```
   [DeviceA-nqa-admin-test1-snmp] destination ip 10.2.2.2
   ```

   # Enable the saving of history records.

   ```
   [DeviceA-nqa-admin-test1-snmp] history-record enable
   [DeviceA-nqa-admin-test1-snmp] quit
   ```

   # Start the SNMP operation.

   ```
   [DeviceA] nqa schedule admin test1 start-time now lifetime forever
   ```

   # After the SNMP operation runs for a period of time, stop the operation.

   ```
   [DeviceA] undo nqa schedule admin test1
   ```

# Display the most recent result of the SNMP operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 50/50/50
    Square-Sum of round trip time: 2500
    Last succeeded probe time: 2011-11-22 10:24:41.1
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

# Display the history records of the SNMP operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          50           Succeeded        2011-11-22 10:24:41.1
```

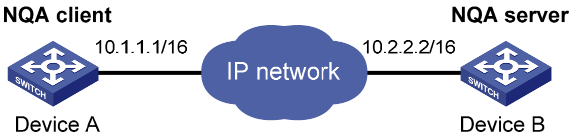
The output shows that it took Device A 50 milliseconds to receive a response from the SNMP agent.

TCP operation configuration example

Network requirements

As shown in Figure 45, configure a TCP operation to test the time required for Device A and Device B to establish a TCP connection.

Figure 45 Network diagram



Configuration procedure

1. Assign each interface an IP address. (Details not shown.)
2. Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)
3. Configure Device B:

   # Enable the NQA server.

   ```
   <DeviceB> system-view
   [DeviceB] nqa server enable
   ```

   # Configure a listening service to listen on the IP address 10.2.2.2 and TCP port 9000.

   ```
   [DeviceB] nqa server tcp-connect 10.2.2.2 9000
   ```

4. Configure Device A:

   # Create a TCP operation.

   ```
   <DeviceA> system-view
   [DeviceA] nqa entry admin test1
   ```

```
[DeviceA-nqa-admin-test1] type tcp
```

# Configure 10.2.2.2 as the destination IP address and port 9000 as the destination port.

```
[DeviceA-nqa-admin-test1-tcp] destination ip 10.2.2.2
[DeviceA-nqa-admin-test1-tcp] destination port 9000
```

# Enable the saving of history records.

```
[DeviceA-nqa-admin-test1-tcp] history-record enable
[DeviceA-nqa-admin-test1-tcp] quit
```

# Start the TCP operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

# After the TCP operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

# Display the most recent result of the TCP operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 13/13/13
    Square-Sum of round trip time: 169
    Last succeeded probe time: 2011-11-22 10:27:25.1
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to disconnect: 0
    Failures due to no connection: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

# Display the history records of the TCP operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          13           Succeeded        2011-11-22 10:27:25.1
```

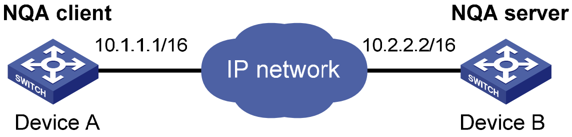
The output shows that it took Device A 13 milliseconds to establish a TCP connection to port 9000 on the NQA server.

UDP echo operation configuration example

Network requirements

As shown in Figure 46, configure a UDP echo operation to test the round-trip time between Device A and Device B. The destination port number is 8000.

Figure 46 Network diagram



Configuration procedure

1. Assign each interface an IP address. (Details not shown.)
2. Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)
3. Configure Device B:

   # Enable the NQA server.

   ```
   <DeviceB> system-view
   [DeviceB] nqa server enable
   ```

   # Configure a listening service to listen on the IP address 10.2.2.2 and UDP port 8000.

   ```
   [DeviceB] nqa server udp-echo 10.2.2.2 8000
   ```

4. Configure Device A:

   # Create a UDP echo operation.

   ```
   <DeviceA> system-view
   [DeviceA] nqa entry admin test1
   [DeviceA-nqa-admin-test1] type udp-echo
   ```

   # Configure 10.2.2.2 as the destination IP address and port 8000 as the destination port.

   ```
   [DeviceA-nqa-admin-test1-udp-echo] destination ip 10.2.2.2
   [DeviceA-nqa-admin-test1-udp-echo] destination port 8000
   ```

   # Enable the saving of history records.

   ```
   [DeviceA-nqa-admin-test1-udp-echo] history-record enable
   [DeviceA-nqa-admin-test1-udp-echo] quit
   ```

   # Start the UDP echo operation.

   ```
   [DeviceA] nqa schedule admin test1 start-time now lifetime forever
   ```

   # After the UDP echo operation runs for a period of time, stop the operation.

   ```
   [DeviceA] undo nqa schedule admin test1
   ```

   # Display the most recent result of the UDP echo operation.

   ```
   [DeviceA] display nqa result admin test1
   NQA entry (admin admin, tag test1) test results:
       Send operation times: 1              Receive response times: 1
       Min/Max/Average round trip time: 25/25/25
       Square-Sum of round trip time: 625
       Last succeeded probe time: 2011-11-22 10:36:17.9
     Extended results:
       Packet loss ratio: 0%
       Failures due to timeout: 0
       Failures due to internal error: 0
       Failures due to other errors: 0
   ```

   # Display the history records of the UDP echo operation.

   ```
   [DeviceA] display nqa history admin test1
   NQA entry (admin admin, tag test1) history records:
     Index      Response     Status          Time
     1          25           Succeeded       2011-11-22 10:36:17.9
   ```

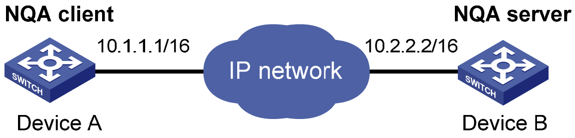
   The output shows that the round-trip time between Device A and port 8000 on Device B is 25 milliseconds.

Voice operation configuration example

Network requirements

As shown in Figure 47, configure a voice operation to test jitters, delay, MOS, and ICPIF between Device A and Device B.

Figure 47 Network diagram



Configuration procedure

1. Assign each interface an IP address. (Details not shown.)
2. Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)
3. Configure Device B:

   # Enable the NQA server.

```
<DeviceB> system-view
[DeviceB] nqa server enable
```

   # Configure a listening service to listen on the IP address 10.2.2.2 and UDP port 9000.

```
[DeviceB] nqa server udp-echo 10.2.2.2 9000
```

4. Configure Device A:

   # Create a voice operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type voice
```

   # Configure 10.2.2.2 as the destination IP address and port 9000 as the destination port.

```
[DeviceA-nqa-admin-test1-voice] destination ip 10.2.2.2
[DeviceA-nqa-admin-test1-voice] destination port 9000
[DeviceA-nqa-admin-test1-voice] quit
```

   # Start the voice operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

   # After the voice operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

   # Display the most recent result of the voice operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1000           Receive response times: 1000
    Min/Max/Average round trip time: 31/1328/33
    Square-Sum of round trip time: 2844813
    Last packet received time: 2011-06-13 09:49:31.1
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
Packets out of sequence: 0
    Packets arrived late: 0
  Voice results:
   RTT number: 1000
    Min positive SD: 1                     Min positive DS: 1
    Max positive SD: 204                   Max positive DS: 1297
    Positive SD number: 257                Positive DS number: 259
    Positive SD sum: 759                   Positive DS sum: 1797
    Positive SD average: 2                 Positive DS average: 6
    Positive SD square-sum: 54127          Positive DS square-sum: 1691967
    Min negative SD: 1                     Min negative DS: 1
    Max negative SD: 203                   Max negative DS: 1297
    Negative SD number: 255                Negative DS number: 259
    Negative SD sum: 759                   Negative DS sum: 1796
    Negative SD average: 2                 Negative DS average: 6
    Negative SD square-sum: 53655          Negative DS square-sum: 1691776
  One way results:
    Max SD delay: 343                      Max DS delay: 985
    Min SD delay: 343                      Min DS delay: 985
    Number of SD delay: 1                  Number of DS delay: 1
    Sum of SD delay: 343                   Sum of DS delay: 985
    Square-Sum of SD delay: 117649         Square-Sum of DS delay: 970225
    SD lost packets: 0                   DS lost packets: 0
    Lost packets for unknown reason: 0
  Voice scores:
    MOS value: 4.38                        ICPIF value: 0
```

# Display the statistics of the voice operation.

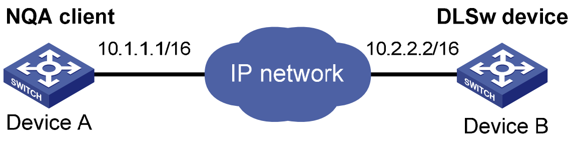
```
[DeviceA] display nqa statistics admin test1
NQA entry (admin admin, tag test1) test statistics:
  NO. : 1

    Start time: 2011-06-13 09:45:37.8
    Life time: 331 seconds
    Send operation times: 4000           Receive response times: 4000
    Min/Max/Average round trip time: 15/1328/32
    Square-Sum of round trip time: 7160528
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
Packets out of sequence: 0
    Packets arrived late: 0
  Voice results:
   RTT number: 4000
    Min positive SD: 1                     Min positive DS: 1
    Max positive SD: 360                   Max positive DS: 1297
    Positive SD number: 1030               Positive DS number: 1024
    Positive SD sum: 4363                  Positive DS sum: 5423
    Positive SD average: 4                 Positive DS average: 5
    Positive SD square-sum: 497725         Positive DS square-sum: 2254957
    Min negative SD: 1                     Min negative DS: 1
    Max negative SD: 360                   Max negative DS: 1297
    Negative SD number: 1028               Negative DS number: 1022
    Negative SD sum: 1028                  Negative DS sum: 1022
    Negative SD average: 4                 Negative DS average: 5
    Negative SD square-sum: 495901         Negative DS square-sum: 5419
  One way results:
    Max SD delay: 359                      Max DS delay: 985
    Min SD delay: 0                        Min DS delay: 0
    Number of SD delay: 4                  Number of DS delay: 4
    Sum of SD delay: 1390                  Sum of DS delay: 1079
    Square-Sum of SD delay: 483202         Square-Sum of DS delay: 973651
    SD lost packets: 0                   DS lost packets: 0
    Lost packets for unknown reason: 0
  Voice scores:
    Max MOS value: 4.38                    Min MOS value: 4.38
    Max ICPIF value: 0                     Min ICPIF value: 0
```

DLSw operation configuration example

Network requirements

As shown in Figure 48, configure a DLSw operation to test the response time of the DLSw device.

Figure 48 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create a DLSw operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type dlsw
```

# Configure 10.2.2.2 as the destination IP address.

```
[DeviceA-nqa-admin-test1-dlsw] destination ip 10.2.2.2
```

# Enable the saving of history records.

```
[DeviceA-nqa-admin-test1-dlsw] history-record enable
```

```
[DeviceA-nqa-admin-test1-dlsw] quit
```

# Start the DLSw operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

# After the DLSw operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

# Display the most recent result of the DLSw operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 19/19/19
    Square-Sum of round trip time: 361
    Last succeeded probe time: 2011-11-22 10:40:27.7
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to disconnect: 0
    Failures due to no connection: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

# Display the history records of the DLSw operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          19           Succeeded        2011-11-22 10:40:27.7
```

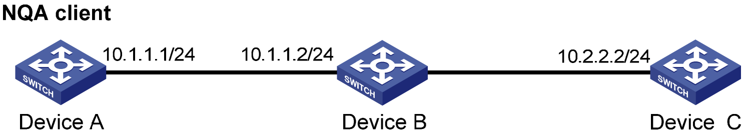
The output shows that the response time of the DLSw device is 19 milliseconds.

Path jitter operation configuration example

Network requirements

As shown in Figure 49, configure a path jitter operation to test the round trip time and jitters from Device A to Device B and Device C.

Figure 49 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Use the **ip ttl-expires enable** command on Device B and use the **ip unreachables enable** command on Device C.

# Create a path jitter operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type path-jitter
```

# Specify 10.2.2.2 as the destination IP address of ICMP echo requests.

```
[DeviceA-nqa-admin-test1-path-jitter] destination ip 10.2.2.2
```

# Configure the path jitter operation to repeat at an interval of 10000 milliseconds.

```
[DeviceA-nqa-admin-test1-path-jitter] frequency 10000
[DeviceA-nqa-admin-test1-path-jitter] quit
```

# Start the path jitter operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

# After the path jitter operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

# Display the most recent result of the path jitter operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
  Hop IP 10.1.1.2
    Basic Results
      Send operation times: 10             Receive response times: 10
      Min/Max/Average round trip time: 9/21/14
      Square-Sum of round trip time: 2419
    Extended Results
      Failures due to timeout: 0
      Failures due to internal error: 0
      Failures due to other errors: 0
      Packets out of sequence: 0
      Packets arrived late: 0
    Path-Jitter Results
      Jitter number: 9
        Min/Max/Average jitter: 1/10/4
      Positive jitter number: 6
        Min/Max/Average positive jitter: 1/9/4
        Sum/Square-Sum positive jitter: 25/173
      Negative jitter number: 3
        Min/Max/Average negative jitter: 2/10/6
        Sum/Square-Sum positive jitter: 19/153

  Hop IP 10.2.2.2
    Basic Results
      Send operation times: 10             Receive response times: 10
      Min/Max/Average round trip time: 15/40/28
      Square-Sum of round trip time: 4493
    Extended Results
      Failures due to timeout: 0
      Failures due to internal error: 0
      Failures due to other errors: 0
      Packets out of sequence: 0
      Packets arrived late: 0
```

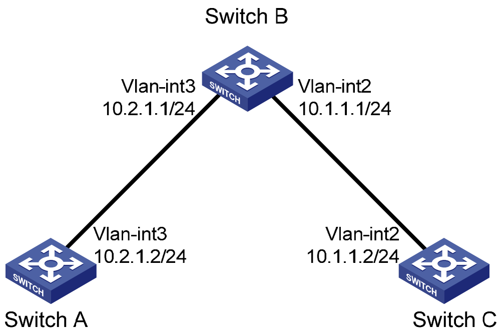
```
Path-Jitter Results
  Jitter number: 9
    Min/Max/Average jitter: 1/10/4
  Positive jitter number: 6
    Min/Max/Average positive jitter: 1/9/4
    Sum/Square-Sum positive jitter: 25/173
  Negative jitter number: 3
    Min/Max/Average negative jitter: 2/10/6
    Sum/Square-Sum positive jitter: 19/153
```

NQA collaboration configuration example

Network requirements

As shown in Figure 50, configure a static route to Switch C with Switch B as the next hop on Switch A. Associate the static route, a track entry, and an ICMP echo operation to monitor the state of the static route.

Figure 50 Network diagram



Configuration procedure

1. Assign each interface an IP address. (Details not shown.)
2. On Switch A, configure a static route, and associate the static route with track entry 1.
```
<SwitchA> system-view
[SwitchA] ip route-static 10.1.1.2 24 10.2.1.1 track 1
```
3. On Switch A, configure an ICMP echo operation:

# Create an NQA operation with the administrator name **admin** and operation tag **test1**.
```
[SwitchA] nqa entry admin test1
```
# Configure the NQA operation type as ICMP echo.
```
[SwitchA-nqa-admin-test1] type icmp-echo
```
# Configure 10.2.1.1 as the destination IP address.
```
[SwitchA-nqa-admin-test1-icmp-echo] destination ip 10.2.1.1
```
# Configure the operation to repeat at an interval of 100 milliseconds.
```
[SwitchA-nqa-admin-test1-icmp-echo] frequency 100
```
# Create reaction entry 1. If the number of consecutive probe failures reaches 5, collaboration is triggered.
```
[SwitchA-nqa-admin-test1-icmp-echo] reaction 1 checked-element probe-fail
threshold-type consecutive 5 action-type trigger-only
```

```
[SwitchA-nqa-admin-test1-icmp-echo] quit
```

# Start the ICMP operation.

```
[SwitchA] nqa schedule admin test1 start-time now lifetime forever
```

4. On Switch A, create track entry 1, and associate it with reaction entry 1 of the NQA operation.

```
[SwitchA] track 1 nqa entry admin test1 reaction 1
```

Verifying the configuration

# Display information about all the track entries on Switch A.

```
[SwitchA] display track all
Track ID: 1
  State: Positive
  Duration: 0 days 0 hours 0 minutes 0 seconds
  Notification delay: Positive 0, Negative 0 (in seconds)
  Tracked object:
    NQA entry: admin test1
    Reaction: 1
```

# Display brief information about active routes in the routing table on Switch A.

```
[SwitchA] display ip routing-table

Destinations : 13        Routes : 13

Destination/Mask    Proto  Pre  Cost         NextHop         Interface
0.0.0.0/32          Direct 0    0            127.0.0.1       InLoop0
10.1.1.0/24         Static 60   0            10.2.1.1        Vlan3
10.2.1.0/24         Direct 0    0            10.2.1.2        Vlan3
10.2.1.0/32         Direct 0    0            10.2.1.2        Vlan3
10.2.1.2/32         Direct 0    0            127.0.0.1       InLoop0
10.2.1.255/32       Direct 0    0            10.2.1.2        Vlan3
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.0/32        Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
127.255.255.255/32  Direct 0    0            127.0.0.1       InLoop0
224.0.0.0/4         Direct 0    0            0.0.0.0         NULL0
224.0.0.0/24        Direct 0    0            0.0.0.0         NULL0
255.255.255.255/32  Direct 0    0            127.0.0.1       InLoop0
```

The output shows that the static route with the next hop 10.2.1.1 is active, and the status of the track entry is positive.

# Remove the IP address of VLAN-interface 3 on Switch B.

```
<SwitchB> system-view
[SwitchB] interface vlan-interface 3
[SwitchB-Vlan-interface3] undo ip address
```

# Display information about all the track entries on Switch A.

```
[SwitchA] display track all
Track ID: 1
  State: Negative
  Duration: 0 days 0 hours 0 minutes 0 seconds
  Notification delay: Positive 0, Negative 0 (in seconds)
  Tracked object:
    NQA entry: admin test1
    Reaction: 1
```

# Display brief information about active routes in the routing table on Switch A.

```
[SwitchA] display ip routing-table

Destinations : 12        Routes : 12

Destination/Mask    Proto  Pre  Cost         NextHop         Interface
0.0.0.0/32          Direct 0    0            127.0.0.1       InLoop0
10.2.1.0/24         Direct 0    0            10.2.1.2        Vlan3
10.2.1.0/32         Direct 0    0            10.2.1.2        Vlan3
10.2.1.2/32         Direct 0    0            127.0.0.1       InLoop0
10.2.1.255/32       Direct 0    0            10.2.1.2        Vlan3
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.0/32        Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
127.255.255.255/32  Direct 0    0            127.0.0.1       InLoop0
224.0.0.0/4         Direct 0    0            0.0.0.0         NULL0
224.0.0.0/24        Direct 0    0            0.0.0.0         NULL0
255.255.255.255/32  Direct 0    0            127.0.0.1       InLoop0
```

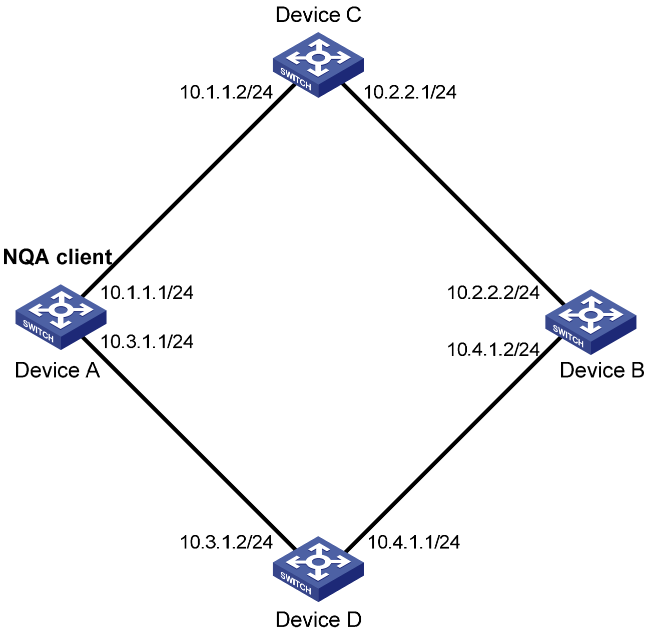
The output shows that the static route does not exist, and the status of the track entry is negative.

ICMP template configuration example

Network requirements

As shown in Figure 51, configure an ICMP template for a feature to perform the ICMP echo operation from Device A to Device B.

Figure 51 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create ICMP template **icmp**.

```
<DeviceA> system-view
[DeviceA] nqa template icmp icmp
```

# Specify 10.2.2.2 as the destination IP address of ICMP echo requests.

```
[DeviceA-nqatplt-icmp-icmp] destination ip 10.2.2.2
```

# Set the probe timeout time for the ICMP echo operation to 500 milliseconds.

```
[DeviceA-nqatplt-icmp-icmp] probe timeout 500
```

# Configure the ICMP echo operation to repeat at an interval of 3000 milliseconds.

```
[DeviceA-nqatplt-icmp-icmp] frequency 3000
```

# If the number of consecutive successful probes reaches 2, the operation succeeds. The NQA client notifies the feature of the successful operation event.

```
[DeviceA-nqatplt-icmp-icmp] reaction trigger probe-pass 2
```

# If the number of consecutive probe failures reaches 2, the operation fails. The NQA client notifies the feature of the operation failure.

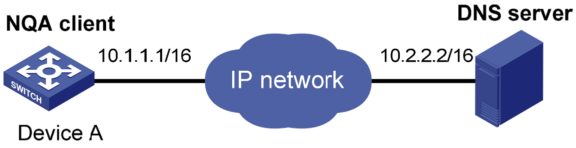
```
[DeviceA-nqatplt-icmp-icmp] reaction trigger probe-fail 2
```

DNS template configuration example

Network requirements

As shown in Figure 52, configure a DNS template for a feature to perform the DNS operation. The operation tests whether Device A can perform the address resolution through the DNS server.

Figure 52 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create DNS template **dns**.

```
<DeviceA> system-view
[DeviceA] nqa template dns dns
```

# Specify the IP address of the DNS server 10.2.2.2 as the destination IP address.

```
[DeviceA-nqatplt-dns-dns] destination ip 10.2.2.2
```

# Specify the domain name to be translated as **host.com**.

```
[DeviceA-nqatplt-dns-dns] resolve-target host.com
```

# Specify the domain name resolution type as type A.

```
[DeviceA-nqatplt-dns-dns] resolve-type A
```

# Specify the expected IP address as 3.3.3.3.

```
[DeviceA-nqatplt-dns-dns] expect ip 3.3.3.3
```

# If the number of consecutive successful probes reaches 2, the operation succeeds. The NQA client notifies the feature of the successful operation event.

```
[DeviceA-nqatplt-dns-dns] reaction trigger probe-pass 2
```

# If the number of consecutive probe failures reaches 2, the operation fails. The NQA client notifies the feature of the operation failure.

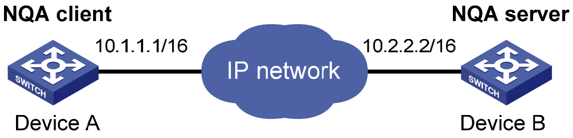
```
[DeviceA-nqatplt-dns-dns] reaction trigger probe-fail 2
```

TCP template configuration example

Network requirements

As shown in Figure 53, configure a TCP template for a feature to perform the TCP operation. The operation tests whether Device A can establish a TCP connection to Device B.

Figure 53 Network diagram



Configuration procedure

1. Assign each interface an IP address. (Details not shown.)

2. Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)
3. Configure Device B:

   # Enable the NQA server.

   ```
   <DeviceB> system-view
   [DeviceB] nqa server enable
   ```

   # Configure a listening service to listen to the IP address 10.2.2.2 and TCP port 9000.

   ```
   [DeviceB] nqa server tcp-connect 10.2.2.2 9000
   ```
4. Configure Device A:

   # Create TCP template **tcp**.

   ```
   <DeviceA> system-view
   [DeviceA] nqa template tcp tcp
   ```

   # Configure 10.2.2.2 as the destination IP address and port 9000 as the destination port.

   ```
   [DeviceA-nqatplt-tcp-tcp] destination ip 10.2.2.2
   [DeviceA-nqatplt-tcp-tcp] destination port 9000
   ```

   # If the number of consecutive successful probes reaches 2, the operation succeeds. The NQA client notifies the feature of the successful operation event.

   ```
   [DeviceA-nqatplt-tcp-tcp] reaction trigger probe-pass 2
   ```

   # If the number of consecutive probe failures reaches 2, the operation fails. The NQA client notifies the feature of the operation failure.

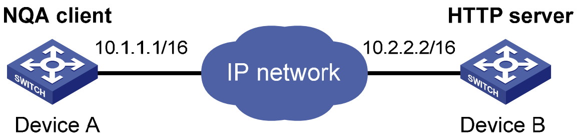
   ```
   [DeviceA-nqatplt-tcp-tcp] reaction trigger probe-fail 2
   ```

HTTP template configuration example

Network requirements

As shown in Figure 54, configure an HTTP template for a feature to perform the HTTP operation. The operation tests whether the NQA client can get data from the HTTP server.

Figure 54 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create HTTP template **http**.

```
<DeviceA> system-view
[DeviceA] nqa template http http
```

# Specify the URL of the server.

```
[DeviceA-nqatplt-http-http] url http://10.2.2.2/index.htm
```

# Configure the HTTP operation to get data from the HTTP server.

```
[DeviceA-nqatplt-http-http] operation get
```

# If the number of consecutive successful probes reaches 2, the operation succeeds. The NQA client notifies the feature of the successful operation event.

```
[DeviceA-nqatplt-http-http] reaction trigger probe-pass 2
```

# If the number of consecutive probe failures reaches 2, the operation fails. The NQA client notifies the feature of the operation failure.

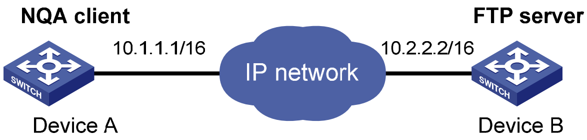
```
[DeviceA-nqatplt-http-http] reaction trigger probe-fail 2
```

FTP template configuration example

Network requirements

As shown in Figure 55, configure an FTP template for a feature to perform the FTP operation. The operation tests whether Device A can upload a file to the FTP server. The login username and password are **admin** and **systemtest**, respectively. The file to be transferred to the FTP server is **config.txt**.

Figure 55 Network diagram



Configuration procedure

# Assign each interface an IP address. (Details not shown.)

# Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

# Create FTP template **ftp**.

```
<DeviceA> system-view
[DeviceA] nqa template ftp ftp
```

# Specify the URL of the FTP server.

```
[DeviceA-nqatplt-ftp-ftp] url ftp://10.2.2.2
```

# Specify 10.1.1.1 as the source IP address.

```
[DeviceA-nqatplt-ftp-ftp] source ip 10.1.1.1
```

# Configure the device to upload file **config.txt** to the FTP server.

```
[DeviceA-nqatplt-ftp-ftp] operation put
[DeviceA-nqatplt-ftp-ftp] filename config.txt
```

# Specify the username for the FTP server login as **admin**.

```
[DeviceA-nqatplt-ftp-ftp] username admin
```

# Specify the password for the FTP server login as **systemtest**.

```
[DeviceA-nqatplt-ftp-ftp] password simple systemtest
```

# If the number of consecutive successful probes reaches 2, the operation succeeds. The NQA client notifies the feature of the successful operation event.

```
[DeviceA-nqatplt-ftp-ftp] reaction trigger probe-pass 2
```

# If the number of consecutive probe failures reaches 2, the operation fails. The NQA client notifies the feature of the operation failure.

```
[DeviceA-nqatplt-ftp-ftp] reaction trigger probe-fail 2
```

Configuring port mirroring

The port mirroring feature is available on Layer 2 Ethernet interfaces.

Overview

Port mirroring copies the packets passing through a port to the monitor port connecting to a data monitoring device for packet analysis.

Terminology

The following terms are used in port mirroring configuration.

Mirroring source

The mirroring sources can be one or more monitored ports, which are called source ports. Packets passing through mirroring sources are copied to a port connecting to a data monitoring device for packet analysis. The copies are called mirrored packets.

Source device

The device where the mirroring sources reside is called a source device.

Mirroring destination

The mirroring destination connects to a data monitoring device and is the destination port (also known as the monitor port) of mirrored packets. Mirrored packets are sent out of the monitor port to the data monitoring device.

A monitor port might receive multiple copies of a packet when it monitors multiple mirroring sources. For example, two copies of a packet are received on Port 1 when the following conditions exist:

- Port 1 is monitoring bidirectional traffic of Port 2 and Port 3 on the same device.
- The packet travels from Port 2 to Port 3.

Destination device

The device where the monitor port resides is called the destination device.

Mirroring direction

The mirroring direction specifies the direction of the traffic that is copied on a mirroring source.

- **Inbound**—Copies packets received.
- **Outbound**—Copies packets sent.
- **Bidirectional**—Copies packets received and sent.

Mirroring group

Port mirroring is implemented through mirroring groups, which include local, remote source, and remote destination groups. For more information about the mirroring groups, see "Port mirroring classification and implementation."

Reflector port, egress port, and remote probe VLAN

Reflector ports, remote probe VLANs, and egress ports are used for Layer 2 remote port mirroring. The remote probe VLAN specially transmits mirrored packets to the destination device. Both the reflector port and egress port reside on a source device and send mirrored packets to the remote probe VLAN. For more information about the reflector port, egress port, remote probe VLAN, and Layer 2 remote port mirroring, see "Port mirroring classification and implementation."

NOTE:

On port mirroring devices, all ports except source, destination, reflector, and egress ports are called common ports.

Port mirroring classification and implementation

Port mirroring includes local port mirroring and remote port mirroring.

- **Local port mirroring**—The mirroring sources and the mirroring destination are on the same device.
- **Remote port mirroring**—The mirroring sources and the mirroring destination are on different devices.

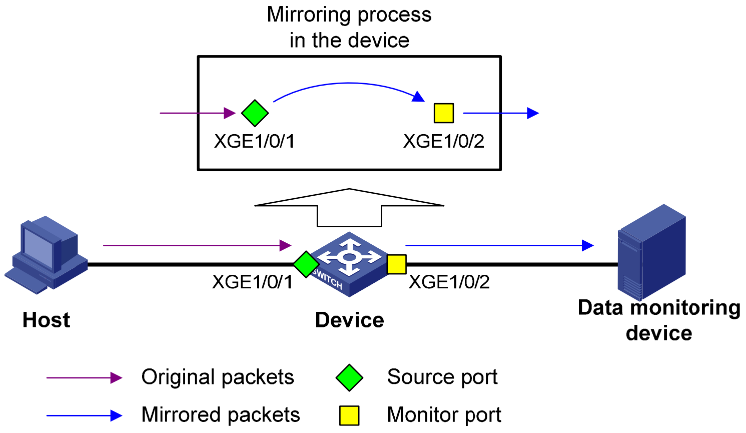
Local port mirroring

In local port mirroring, the following conditions exist:

- The source device is directly connected to a data monitoring device.
- The source device acts as the destination device to forward mirrored packets to the data monitoring device.

A local mirroring group is a mirroring group that contains the mirroring source and the mirroring destination on the same device.

Figure 56 Local port mirroring implementation



As shown in Figure 56, the source port Ten-GigabitEthernet 1/0/1 and the monitor port Ten-GigabitEthernet 1/0/2 reside on the same device. Packets received on Ten-GigabitEthernet 1/0/1 are copied to Ten-GigabitEthernet 1/0/2. Ten-GigabitEthernet 1/0/2 then forwards the packets to the data monitoring device for analysis.

Remote port mirroring

In remote port mirroring, the following conditions exist:

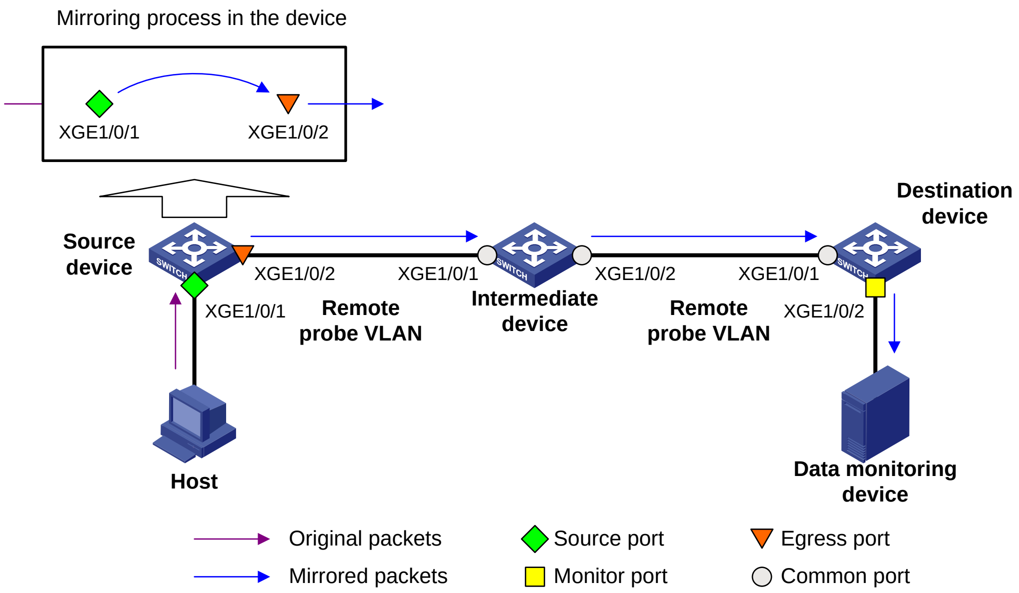
- The source device is not directly connected to a data monitoring device.
- The source device copies mirrored packets to the destination device, which forwards them to the data monitoring device.
- The mirroring sources and the mirroring destination reside on different devices and are in different mirroring groups.

A remote source group or remote destination group is a mirroring group that contains the mirroring sources or the mirroring destination, respectively. Intermediate devices are the devices between the source devices and the destination device.

In Layer 2 remote port mirroring, the mirroring source and the mirroring destination are located on different devices on a same Layer 2 network.

The source device copies packets received on the source port to the egress port. The egress port forwards the packets to the intermediate devices. The intermediate devices then flood the packets in the remote probe VLAN and transmit the mirrored packets to the destination device. Upon receiving the mirrored packets, the destination device determines whether the ID of the mirrored packets is the same as the remote probe VLAN ID. If the two VLAN IDs match, the device forwards the mirrored packets to the data monitoring device through the monitor port.

Figure 57 Layer 2 remote port mirroring implementation



To ensure Layer 2 forwarding of the mirrored packets, assign the intermediate devices' ports facing the source and destination devices to the remote probe VLAN.

To monitor the bidirectional traffic of a port in a mirroring group, disable MAC address learning for the remote probe VLAN on the source, intermediate, and destination devices. For more information about MAC address learning, see *Layer 2—LAN Switching Configuration Guide*.

Configuring local port mirroring

A local mirroring group takes effect only when you configure the source ports and the monitor port for the local mirroring group.

Local port mirroring configuration task list

Tasks at a glance
1. (Required.) Creating a local mirroring group
2. (Required.) Configuring source ports for the local mirroring group
3. (Required.) Configuring the monitor port for the local mirroring group

Creating a local mirroring group

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a local mirroring group.	**mirroring-group** *group-id* **local**	By default, no local mirroring group exists.

Configuring source ports for the local mirroring group

To configure source ports for a local mirroring group, use one of the following methods:

- Assign a list of source ports to the mirroring group in system view.
- Assign a port to the mirroring group as a source port in interface view.

  To assign multiple ports to the mirroring group as source ports in interface view, repeat the operation.

Configuration restrictions and guidelines

When you configure source ports for a local mirroring group, follow these restrictions and guidelines:

- A mirroring group can contain multiple source ports.
- A port can act as a source port for multiple mirroring groups.
- A source port cannot be used as a reflector port, egress port, or monitor port.
- When you configure a TRILL access port as a source port, only non-TRILL-encapsulated packets can be mirrored. Other packets are dropped.
- When you configure a TRILL trunk port as a source port, only TRILL-encapsulated packets can be mirrored. Other packets are dropped.
- When you configure a TRILL hybrid port as a source port, both TRILL-encapsulated and non-TRILL-encapsulated packets can be mirrored.

Configuring source ports in system view

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure source ports for the specified local mirroring group.	**mirroring-group** *group-id* **mirroring-port** *interface-list* { **both** | **inbound** | **outbound** }	By default, no source port is configured for a local mirroring group.

Configuring source ports in interface view

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Configure the port as a source port for the specified local mirroring group.	**mirroring-group** *group-id* **mirroring-port** { **both** | **inbound** | **outbound** }	By default, a port does not act as a source port for any local mirroring group.

Configuring the monitor port for the local mirroring group

To configure the monitor port for a mirroring group, use one of the following methods:

- Configure the monitor port for the mirroring group in system view.
- Assign a port to the mirroring group as the monitor port in interface view.

Configuration restrictions and guidelines

When you configure the monitor port for a local mirroring group, follow these restrictions and guidelines:

- Do not enable the spanning tree feature on the monitor port.
- For a Layer 2 aggregate interface configured as the monitor port, do not configure its member ports as source ports.
- A mirroring group contains only one monitor port.
- Use a monitor port only for port mirroring, so the data monitoring device receives only the mirrored traffic.

Configuring the monitor port in system view

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure the monitor port for the specified local mirroring group.	**mirroring-group** *group-id* **monitor-port** *interface-type interface-number*	By default, no monitor port is configured for a local mirroring group.

Configuring the monitor port in interface view

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Configure the port as the monitor port for the specified mirroring group.	**mirroring-group** *group-id* **monitor-port**	By default, a port does not act as the monitor port for any local mirroring group.

Configure local port mirroring with multiple monitor ports

Typically, you can configure only one monitor port in a local mirroring group. To configure local port mirroring to support multiple monitor ports, use the remote probe VLAN.

In Layer 2 remote port mirroring, mirrored packets are broadcast within the remote probe VLAN.

To broadcast mirrored packets to multiple local monitor ports through the remote probe VLAN, perform the following tasks:

1. Create a remote source group on the local device.
2. Specify the reflector port for this mirroring group.
3. Configure the remote probe VLAN for this mirroring group.
4. Assign the local monitor ports to the remote probe VLAN.

Configuration restrictions and guidelines

When you configure local port mirroring with multiple monitor ports, follow these restrictions and guidelines:

- Configure an unused port on the device as the reflector port. Do not connect a network cable to the reflector port.
- When a port is configured as a reflector port, the port restores to the factory default settings. Do not configure other features on the reflector port.
- A mirroring group can contain multiple source ports.
- For correct operation of port mirroring, do not assign a source port to the remote probe VLAN.
- If you have configured a reflector port for a remote source group, do not configure an egress port for it.
- A VLAN can act as the remote probe VLAN for only one remote source group. HP recommends that you use the remote probe VLAN for port mirroring exclusively. Do not create a VLAN interface or perform other configurations for the VLAN.
- A remote probe VLAN must be a static VLAN. To delete this static VLAN, first remove the remote probe VLAN configuration by using the **undo mirroring-group remote-probe vlan** command.
- If the remote probe VLAN of a remote mirroring group is removed, the remote mirroring group will become invalid.

Configuration procedure

To configure local port mirroring with multiple monitor ports:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a remote source group.	**mirroring-group** *group-id* **remote-source**	By default, no mirroring groups exist on a device.

Step	Command	Remarks
3. Configure source ports for the remote source group.	• In system view: **mirroring-group** *group-id* **mirroring-port** *mirroring-port-list* { **both** | **inbound** | **outbound** } • In interface view: a. Enter interface view: **interface** *interface-type interface-number* b. Assign the port to the mirroring group as a source port: **mirroring-group** *group-id* **mirroring-port** { **both** | **inbound** | **outbound** } c. Return to system view: **quit**	By default, no source port is configured for a mirroring group.
4. Configure the reflector port for the remote source group.	**mirroring-group** *group-id* **reflector-port** *reflector-port*	By default, no reflector port is configured for a mirroring group.
5. Create a VLAN to be configured as the remote probe VLAN.	**vlan** *vlan-id*	By default, only VLAN 1 (system default VLAN) exists.
6. Assign monitor ports to the VLAN.	**port** *interface-list*	By default, all ports are in VLAN 1.
7. Return to system view.	**quit**	N/A
8. Configure the VLAN above as the remote probe VLAN for the remote source group.	**mirroring-group** *group-id* **remote-probe vlan** *rprobe-vlan-id*	By default, no remote probe VLAN is configured for a mirroring group.

Configuring Layer 2 remote port mirroring

To configure Layer 2 remote port mirroring, perform the following tasks:

- Configure a remote source group on the source device.
- Configure a cooperating remote destination group on the destination device.
- If intermediate devices exist, configure the following devices and ports to allow the remote probe VLAN to pass through.
    - Intermediate devices.
    - Ports connected to the intermediate devices on the source and destinations devices.

When you configure Layer 2 remote port mirroring, follow these guidelines:

- For a mirrored packet to successfully arrive at the remote destination device, make sure its VLAN ID is not removed or changed.
- The switch does not support configuring Layer 2 aggregate interfaces as source ports or monitor ports for Layer 2 remote port mirroring.
- Do not enable MVRP on the devices or ports that allow the remote probe VLAN to pass through. If MVRP is enabled, MVRP might register the remote probe VLAN with unexpected ports, resulting in

undesired copies. For more information about MVRP, see *Layer 2—LAN Switching Configuration Guide*.

- HP recommends that you configure devices in the order of the destination device, the intermediate devices, and the source device.

Tasks at a glance
(Required.) Configuring a remote destination group on the destination device: 1. Creating a remote destination group 2. Configuring the monitor port for a remote destination group 3. Configuring the remote probe VLAN for a remote destination group 4. Assigning the monitor port to the remote probe VLAN
(Required.) Configuring a remote source group on the source device: 1. Creating a remote source group 2. Configuring source ports for a remote source group 3. Configuring the egress port for a remote source group 4. Configuring the remote probe VLAN for a remote source group

Configuring a remote destination group on the destination device

Creating a remote destination group

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a remote destination group.	**mirroring-group** *group-id* **remote-destination**	By default, no remote destination group exists on a device.

Configuring the monitor port for a remote destination group

To configure the monitor port for a mirroring group, use one of the following methods:

- Configure the monitor port for the mirroring group in system view.
- Assign a port to the mirroring group as the monitor port in interface view.

When you configure the monitor port for a remote destination group, follow these restrictions and guidelines:

- Do not enable the spanning tree feature on the monitor port.
- Use a monitor port only for port mirroring, so the data monitoring device receives only the mirrored traffic.
- A mirroring group must contain only one monitor port.
- A monitor port can belong to only one mirroring group.

Configuring the monitor port for a remote destination group in system view

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step		Command	Remarks
2.	Configure the monitor port for the specified remote destination group.	**mirroring-group** *group-id* **monitor-port** *interface-type interface-number*	By default, no monitor port is configured for a remote destination group.

Configuring the monitor port for a remote destination group in interface view

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the port as the monitor port for the specified remote destination group.	**mirroring-group** *group-id* **monitor-port**	By default, a port does not act as the monitor port for any remote destination group.

Configuring the remote probe VLAN for a remote destination group

When you configure the remote probe VLAN for a remote destination group, follow these restrictions and guidelines:

- Only an existing static VLAN can be configured as a remote probe VLAN.
- When a VLAN is configured as a remote probe VLAN, use the remote probe VLAN for port mirroring exclusively.
- Configure the same remote probe VLAN for the remote groups on the source and destination devices.

To configure the remote probe VLAN for a remote destination group:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Configure the remote probe VLAN for the specified remote destination group.	**mirroring-group** *group-id* **remote-probe vlan** *vlan-id*	By default, no remote probe VLAN is configured for a remote destination group.

Assigning the monitor port to the remote probe VLAN

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter the interface view of the monitor port.	**interface** *interface-type interface-number*	N/A

Step	Command	Remarks
3. Assign the port to the remote probe VLAN.	- For an access port: **port access vlan** *vlan-id* - For a trunk port: **port trunk permit vlan** *vlan-id* - For a hybrid port: **port hybrid vlan** *vlan-id* { **tagged** | **untagged** }	For more information about the **port access vlan**, **port trunk permit vlan**, and **port hybrid vlan** commands, see *Layer 2—LAN Switching Command Reference*.

Configuring a remote source group on the source device

Creating a remote source group

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a remote source group.	**mirroring-group** *group-id* **remote-source**	By default, no remote source group exists on a device.

Configuring source ports for a remote source group

To configure source ports for a mirroring group, use one of the following methods:

- Assign a list of source ports to the mirroring group in system view.
- Assign a port to the mirroring group as a source port in interface view.

  To assign multiple ports to the remote source group as source ports in interface view, repeat the operation.

When you configure source ports for a remote source group, follow these restrictions and guidelines:

- Do not assign a source port of a remote source group to the remote probe VLAN of the remote source group.
- A mirroring group can contain multiple source ports.
- A port can act as a source port for multiple mirroring groups.
- A source port cannot be used as a reflector port, monitor port, or egress port.
- When you configure a TRILL access port as a source port, only non-TRILL-encapsulated packets can be mirrored. Other packets are dropped.
- When you configure a TRILL trunk port as a source port, only TRILL-encapsulated packets can be mirrored. Other packets are dropped.
- When you configure a TRILL hybrid port as a source port, both TRILL-encapsulated and non-TRILL-encapsulated packets can be mirrored.

Configuring source ports for a remote source group in system view

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step		Command	Remarks
2.	Configure source ports for the specified remote source group.	**mirroring-group** *group-id* **mirroring-port** *interface-list* { **both** | **inbound** | **outbound** }	By default, no source port is configured for a remote source group.

Configuring a source port for a remote source group in interface view

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the port as a source port for the specified remote source group.	**mirroring-group** *group-id* **mirroring-port** { **both** | **inbound** | **outbound** }	By default, a port does not act as a source port for any remote source group.

Configuring the egress port for a remote source group

To configure the egress port for a remote source group, use one of the following methods:

- Configure the egress port for the remote source group in system view.
- Assign a port to the remote source group as the egress port in interface view.

When you configure the egress port for a remote source group, follow these guidelines:

- Disable the following features on the egress port:
  - Spanning tree.
  - 802.1X.
  - IGMP snooping.
  - Static ARP.
  - MAC address learning.
- A mirroring group contains only one egress port.
- A port of an existing mirroring group cannot be configured as an egress port.

Configuring the egress port for a remote source group in system view

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Configure the egress port for the specified remote source group.	**mirroring-group** *group-id* **monitor-egress** *interface-type interface-number*	By default, no egress port is configured for a remote source group.

Configuring the egress port for a remote source group in interface view

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter interface view.	**interface** *interface-type interface-number*	N/A
3.	Configure the port as the egress port for the specified remote source group.	**mirroring-group** *group-id* **monitor-egress**	By default, a port does not act as the egress port for any remote source group.

Configuring the remote probe VLAN for a remote source group

When you configure the remote probe VLAN for a remote source group, follow these restrictions and guidelines:

- Only an existing static VLAN can be configured as a remote probe VLAN.
- When a VLAN is configured as a remote probe VLAN, use the VLAN for port mirroring exclusively.
- The remote mirroring groups on the source device and destination device must use the same remote probe VLAN.

To configure the remote probe VLAN for a remote source group:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Configure the remote probe VLAN for the specified remote source group.	**mirroring-group** *group-id* **remote-probe vlan** *vlan-id*	By default, no remote probe VLAN is configured for a remote source group.

Displaying and maintaining port mirroring

Execute **display** commands in any view.

Task	Command
Display mirroring group information.	**display mirroring-group** { *group-id* | **all** | **local** | **remote-destination** | **remote-source** }

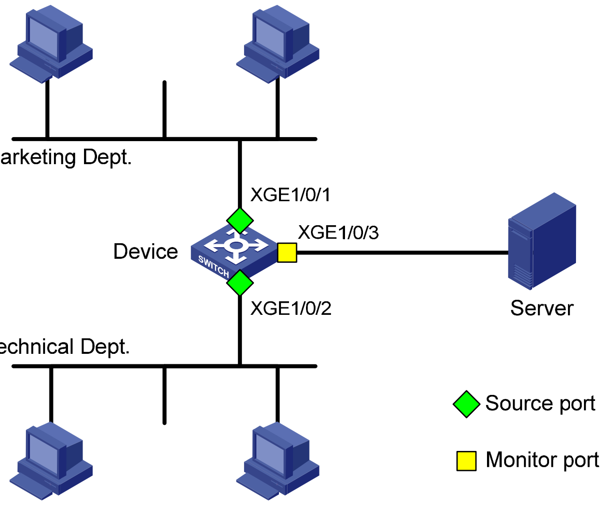
Port mirroring configuration examples

Local port mirroring configuration example

Network requirements

As shown in Figure 58, configure local port mirroring so the server can monitor the bidirectional traffic of the Marketing department and the Technical department.

Figure 58 Network diagram



Configuration procedure

# Create local mirroring group 1.

```
<Device> system-view
[Device] mirroring-group 1 local
```

# Configure Ten-GigabitEthernet 1/0/1 and Ten-GigabitEthernet 1/0/2 as source ports for local mirroring group 1.

```
[Device] mirroring-group 1 mirroring-port ten-gigabitethernet 1/0/1 ten-gigabitethernet
1/0/2 both
```

# Configure Ten-GigabitEthernet 1/0/3 as the monitor port for local mirroring group 1.

```
[Device] mirroring-group 1 monitor-port ten-gigabitethernet 1/0/3
```

# Disable the spanning tree feature on the monitor port Ten-GigabitEthernet 1/0/3.

```
[Device] interface ten-gigabitethernet 1/0/3
[Device-Ten-GigabitEthernet1/0/3] undo stp enable
[Device-Ten-GigabitEthernet1/0/3] quit
```

Verifying the configuration

# Display information about all mirroring groups.

```
[Device] display mirroring-group all
Mirroring group 1:
    Type: Local
    Status: Active
    Mirroring port:
        Ten-GigabitEthernet1/0/1  Both
        Ten-GigabitEthernet1/0/2  Both
    Monitor port: Ten-GigabitEthernet1/0/3
```

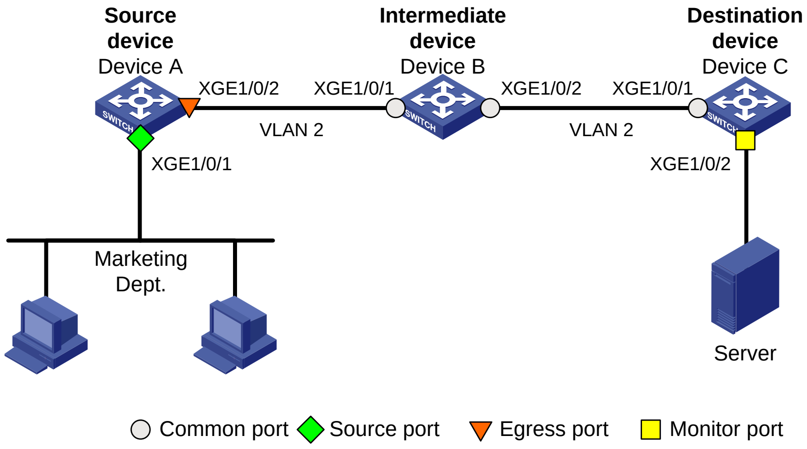
The output shows that you can monitor all packets received and sent by the Marketing department and the Technical department on the server.

Layer 2 remote port mirroring configuration example

Network requirements

As shown in Figure 59, configure Layer 2 remote port mirroring so the server can monitor the bidirectional traffic of the Marketing department.

Figure 59 Network diagram



Configuration procedure

1. Configure Device C (the destination device):

# Configure Ten-GigabitEthernet 1/0/1 as a trunk port to permit the packets from VLAN 2 to pass through.

```
<DeviceC> system-view
[DeviceC] interface ten-gigabitethernet 1/0/1
[DeviceC-Ten-GigabitEthernet1/0/1] port link-type trunk
[DeviceC-Ten-GigabitEthernet1/0/1] port trunk permit vlan 2
[DeviceC-Ten-GigabitEthernet1/0/1] quit
```

# Create a remote destination group.

```
[DeviceC] mirroring-group 2 remote-destination
```

# Create VLAN 2.

```
[DeviceC] vlan 2
```

# Disable MAC address learning for VLAN 2.

```
[DeviceC-vlan2] undo mac-address mac-learning enable
[DeviceC-vlan2] quit
```

# Configure VLAN 2 as the remote probe VLAN and Ten-GigabitEthernet 1/0/2 as the monitor port of the mirroring group.

```
[DeviceC] mirroring-group 2 remote-probe vlan 2
[DeviceC] interface ten-gigabitethernet 1/0/2
[DeviceC-Ten-GigabitEthernet1/0/2] mirroring-group 2 monitor-port
```

# Disable the spanning tree feature on Ten-GigabitEthernet 1/0/2.

```
[DeviceC-Ten-GigabitEthernet1/0/2] undo stp enable
```

# Assign Ten-GigabitEthernet 1/0/2 to VLAN 2 as an access port.

```
[DeviceC-Ten-GigabitEthernet1/0/2] port access vlan 2
```

```
[DeviceC-Ten-GigabitEthernet1/0/2] quit
```

2. Configure Device B (the intermediate device):

   # Create VLAN 2.

   ```
   <DeviceB> system-view
   [DeviceB] vlan 2
   ```

   # Disable MAC address learning for VLAN 2.

   ```
   [DeviceB-vlan2] undo mac-address mac-learning enable
   [DeviceB-vlan2] quit
   ```

   # Configure Ten-GigabitEthernet 1/0/1 as a trunk port to permit the packets from VLAN 2 to pass through.

   ```
   [DeviceB] interface ten-gigabitethernet 1/0/1
   [DeviceB-Ten-GigabitEthernet1/0/1] port link-type trunk
   [DeviceB-Ten-GigabitEthernet1/0/1] port trunk permit vlan 2
   [DeviceB-Ten-GigabitEthernet1/0/1] quit
   ```

   # Configure Ten-GigabitEthernet 1/0/2 as a trunk port to permit the packets from VLAN 2 to pass through.

   ```
   [DeviceB] interface ten-gigabitethernet 1/0/2
   [DeviceB-Ten-GigabitEthernet1/0/2] port link-type trunk
   [DeviceB-Ten-GigabitEthernet1/0/2] port trunk permit vlan 2
   [DeviceB-Ten-GigabitEthernet1/0/2] quit
   ```

3. Configure Device A (the source device):

   # Create a remote source group.

   ```
   <DeviceA> system-view
   [DeviceA] mirroring-group 1 remote-source
   ```

   # Create VLAN 2.

   ```
   [DeviceA] vlan 2
   ```

   # Disable MAC address learning for VLAN 2.

   ```
   [DeviceA-vlan2] undo mac-address mac-learning enable
   [DeviceA-vlan2] quit
   ```

   # Configure VLAN 2 as the remote probe VLAN of the mirroring group.

   ```
   [DeviceA] mirroring-group 1 remote-probe vlan 2
   ```

   # Configure Ten-GigabitEthernet 1/0/1 as a source port and Ten-GigabitEthernet 1/0/2 as the egress port in the mirroring group.

   ```
   [DeviceA] mirroring-group 1 mirroring-port ten-gigabitethernet 1/0/1 both
   [DeviceA] mirroring-group 1 monitor-egress ten-gigabitethernet 1/0/2
   ```

   # Configure Ten-GigabitEthernet 1/0/2 as a trunk port to permit the packets from VLAN 2 to pass through.

   ```
   [DeviceA] interface ten-gigabitethernet 1/0/2
   [DeviceA-Ten-GigabitEthernet1/0/2] port link-type trunk
   [DeviceA-Ten-GigabitEthernet1/0/2] port trunk permit vlan 2
   ```

   # Disable the spanning tree feature on Ten-GigabitEthernet 1/0/2.

   ```
   [DeviceA-Ten-GigabitEthernet1/0/2] undo stp enable
   [DeviceA-Ten-GigabitEthernet1/0/2] quit
   ```

Verifying the configuration

# Display information about all mirroring groups on Device C.

```
[DeviceC] display mirroring-group all
Mirroring group 2:
    Type: Remote destination
    Status: Active
    Monitor port: Ten-GigabitEthernet1/0/2
    Remote probe VLAN: 2
```

# Display information about all mirroring groups on Device A.

```
[DeviceA] display mirroring-group all
Mirroring group 1:
    Type: Remote source
    Status: Active
    Mirroring port:
        Ten-GigabitEthernet1/0/1  Both
    Monitor egress port: Ten-GigabitEthernet1/0/2
    Remote probe VLAN: 2
```

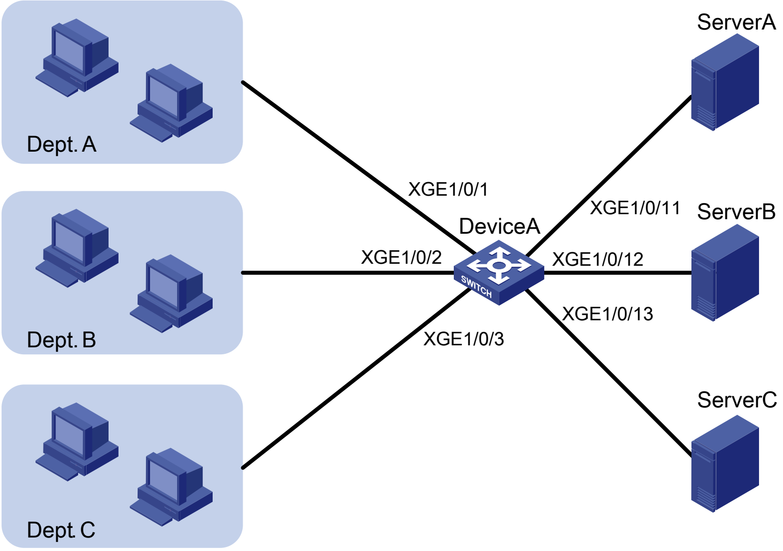
The output shows that you can monitor all packets received and sent by the Marketing department on the server.

Local port mirroring with multiple monitor ports configuration example

Network requirements

As shown in Figure 60, configure port mirroring so servers A, B, and C can monitor the bidirectional traffic of the three departments.

Figure 60 Network diagram



Configuration procedure

# Create remote source group 1.

```
<DeviceA> system-view
[DeviceA] mirroring-group 1 remote-source
```

# Configure Ten-GigabitEthernet 1/0/1 through Ten-GigabitEthernet 1/0/3 as source ports of the remote source group.

```
[DeviceA] mirroring-group 1 mirroring-port ten-gigabitethernet 1/0/1 to
ten-gigabitethernet 1/0/3 both
```

# Configure an unused port (Ten-GigabitEthernet 1/0/5, for example) of Device A as the reflector port of the remote source group.

```
[DeviceA] mirroring-group 1 reflector-port ten-gigabitethernet 1/0/5
This operation may delete all settings made on the interface. Continue? [Y/N]:y
```

# Create VLAN 10, and assign ports Ten-GigabitEthernet 1/0/11 through Ten-GigabitEthernet 1/0/13 to VLAN 10.

```
[DeviceA] vlan 10
[DeviceA-vlan10] port ten-gigabitethernet 1/0/11 to ten-gigabitethernet 1/0/13
[DeviceA-vlan10] quit
```

# Configure VLAN 10 as the remote probe VLAN of the remote source group.

```
[DeviceA] mirroring-group 1 remote-probe vlan 10
```

Configuring flow mirroring

The flow mirroring feature is available on Layer 2 Ethernet interfaces.

Overview

Flow mirroring copies packets matching a class to a destination for analyzing and monitoring. It is implemented through QoS policies.

To configure flow mirroring, perform the following tasks:

- Define traffic classes and configure match criteria to classify packets to be mirrored. Flow mirroring allows you to flexibly classify packets to be analyzed by defining match criteria.
- Configure traffic behaviors to mirror the matching packets to the specified destination.

You can configure an action to mirror matching packets to one of the following destinations:

- **Interface**—The matching packets are copied to an interface connecting to a data monitoring device. The data monitoring device analyzes the packets received on the interface.
- **CPU**—The matching packets are copied to the CPU of the IRF member device where they are received. The CPU analyzes the packets or delivers them to upper layers.

For more information about QoS policies, traffic classes, and traffic behaviors, see *ACL and QoS Configuration Guide*.

Flow mirroring configuration task list

Tasks at a glance
(Required.) Configuring match criteria
(Required.) Configuring a traffic behavior
(Required.) Configuring a QoS policy
(Required.) Applying a QoS policy: • Applying a QoS policy to an interface • Applying a QoS policy to a VLAN • Applying a QoS policy globally • Applying a QoS policy to the control plane

For more information about the following commands except the **mirror-to** command, see *ACL and QoS Command Reference*.

Configuring match criteria

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. Create a class and enter class view.	**traffic classifier** *tcl-name* [**operator** { **and** | **or** }]	By default, no traffic class exists.
3. Configure match criteria.	**if-match** *match-criteria*	By default, no match criterion is configured in a traffic class.

Configuring a traffic behavior

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	By default, no traffic behavior exists.
3. Configure a mirroring action for the traffic behavior.	• Mirror traffic to an interface: **mirror-to interface** *interface-type interface-number* [**destination-ip** *destination-ip-address* **source-ip** *source-ip-address* [**dscp** *dscp-value* | **vlan** *vlan-id*] *] • Mirror traffic to a CPU: **mirror-to cpu**	By default, no mirroring action is configured for a traffic behavior. When the destination IP address is specified for mirrored packets, the output interface of the route to the destination address does not support ECMP.
4. (Optional.) Display traffic behavior configuration.	• **display traffic behavior**	Available in any view.

Configuring a QoS policy

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a QoS policy and enter QoS policy view.	**qos policy** *policy-name*	By default, no QoS policy exists.
3. Associate a class with a traffic behavior in the QoS policy.	**classifier** *tcl-name* **behavior** *behavior-name*	By default, no traffic behavior is associated with a class.
4. (Optional.) Display QoS policy configuration.	**display qos policy**	Available in any view.

Applying a QoS policy

Applying a QoS policy to an interface

By applying a QoS policy to an interface, you can mirror the traffic in the specified direction on the interface. A policy can be applied to multiple interfaces, but in one direction (inbound or outbound) of an interface, only one policy can be applied.

To apply a QoS policy to an interface:

Step	Command
1. Enter system view.	**system-view**
2. Enter interface view.	**interface** *interface-type interface-number*
3. Apply a policy to the interface.	**qos apply policy** *policy-name* { **inbound** | **outbound** }

Applying a QoS policy to a VLAN

You can apply a QoS policy to a VLAN to mirror the traffic in the specified direction on all ports in the VLAN.

To apply the QoS policy to a VLAN:

Step	Command
1. Enter system view.	**system-view**
2. Apply a QoS policy to a VLAN.	**qos vlan-policy** *policy-name* **vlan** *vlan-id-list* { **inbound** | **outbound** }

Applying a QoS policy globally

You can apply a QoS policy globally to mirror the traffic in the specified direction on all ports.

To apply a QoS policy globally:

Step	Command
1. Enter system view.	**system-view**
2. Apply a QoS policy globally.	**qos apply policy** *policy-name* **global** { **inbound** | **outbound** }

Applying a QoS policy to the control plane

You can apply a QoS policy to the control plane to mirror the traffic in the specified direction on all ports of the control plane.

To apply a QoS policy to the control plane:

Step	Command
1. Enter system view.	**system-view**
2. Enter control plane view.	**control-plane slot** *slot-number*
3. Apply a QoS policy to the control plane.	**qos apply policy** *policy-name* **inbound**

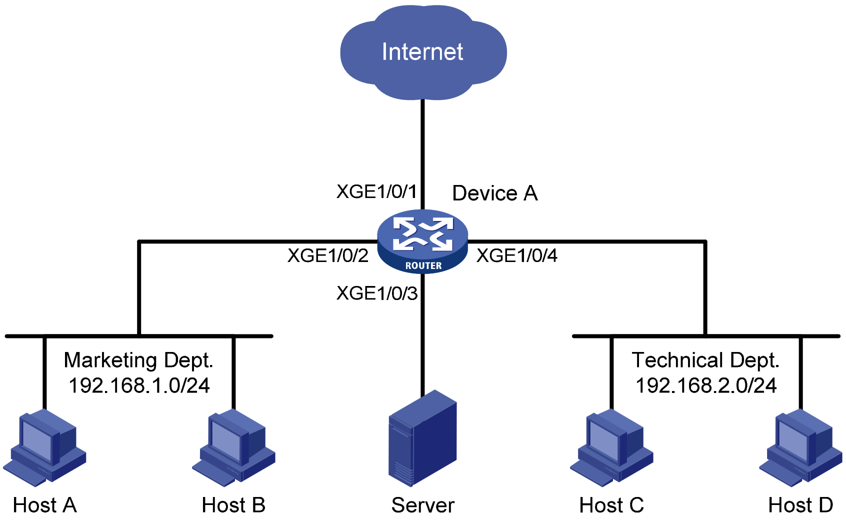
Flow mirroring configuration example

Network requirements

As shown in Figure 61, configure flow mirroring so that the server can monitor following traffic:

- All traffic that the Technical department sends to access the Internet.
- IP traffic that the Technical department sends to the Marketing department during working hours (8:00 to 18:00) on weekdays.

Figure 61 Network diagram



Configuration procedure

# Create a working hour range **work**, in which the working hour is from 8:00 to 18:00 on weekdays.

```
<DeviceA> system-view
[DeviceA] time-range work 8:00 to 18:00 working-day
```

# Create ACL 3000 to allow packets from the Technical department to access the Internet and to the Marketing department during working hours.

```
[DeviceA] acl number 3000
[DeviceA-acl-adv-3000] rule permit tcp source 192.168.2.0 0.0.0.255 destination-port eq
www
[DeviceA-acl-adv-3000] rule permit ip source 192.168.2.0 0.0.0.255 destination
192.168.1.0 0.0.0.255 time-range work
[DeviceA-acl-adv-3000] quit
```

# Create traffic class **tech_c**, and configure the match criterion as ACL 3000.

```
[DeviceA] traffic classifier tech_c
[DeviceA-classifier-tech_c] if-match acl 3000
[DeviceA-classifier-tech_c] quit
```

# Create traffic behavior **tech_b**, configure the action of mirroring traffic to port Ten-GigabitEthernet 1/0/3.

```
[DeviceA] traffic behavior tech_b
[DeviceA-behavior-tech_b] mirror-to interface ten-gigabitethernet 1/0/3
[DeviceA-behavior-tech_b] quit
```

# Create QoS policy **tech_p**, and associate traffic class **tech_c** with traffic behavior **tech_b** in the QoS policy.

```
[DeviceA] qos policy tech_p
[DeviceA-qospolicy-tech_p] classifier tech_c behavior tech_b
[DeviceA-qospolicy-tech_p] quit
```

# Apply QoS policy **tech_p** to the incoming packets of Ten-GigabitEthernet 1/0/4.

```
[DeviceA] interface ten-gigabitethernet 1/0/4
[DeviceA-Ten-GigabitEthernet1/0/4] qos apply policy tech_p inbound
[DeviceA-Ten-GigabitEthernet1/0/4] quit
```

Verifying the configuration

# Verify that the server can monitor the following traffic:

- All traffic sent by the Technical department to access the Internet.
- The IP traffic that the Technical department sends to the Marketing department during working hours on weekdays.

(Details not shown.)

Configuring sFlow

sFlow is a traffic monitoring technology.

As shown in Figure 62, the sFlow system involves an sFlow agent embedded in a device and a remote sFlow collector. The sFlow agent collects interface counter information and packet information and encapsulates the sampled information in sFlow packets. When the sFlow packet buffer is full, or the aging timer (fixed to 1 second) expires, the sFlow agent performs the following tasks:

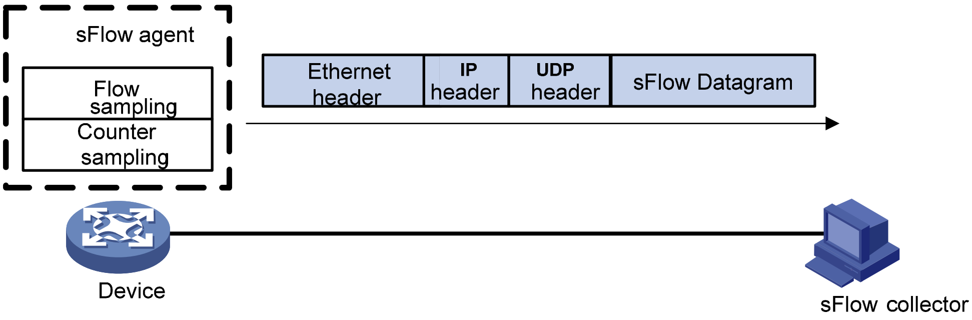
- Encapsulates the sFlow packets in the UDP datagrams.
- Sends the UDP datagrams to the specified sFlow collector.

The sFlow collector analyzes the information and displays the results. One sFlow collector can monitor multiple sFlow agents.

sFlow provides the following sampling mechanisms:

- **Flow sampling**—Obtains packet information.
- **Counter sampling**—Obtains interface counter information.

Figure 62 sFlow system



Protocols and standards

- RFC 3176, *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks*
- sFlow.org, *sFlow Version 5*

sFlow configuration task list

Tasks at a glance
(Required.) Configuring the sFlow agent and sFlow collector information
Perform at least one of the following tasks: • Configuring flow sampling • Configuring counter sampling

Configuring the sFlow agent and sFlow collector information

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. (Optional.) Configure an IP address for the sFlow agent.	**sflow agent** { **ip** *ip-address* | **ipv6** *ipv6-address* }	By default, no IP address is configured for the sFlow agent. The device periodically checks whether the sFlow agent has an IP address. If not, the device automatically selects an IPv4 address for the sFlow agent but does not save the IPv4 address in the configuration file. NOTE: • HP recommends that you manually configure an IP address for the sFlow agent. • Only one IP address can be configured for the sFlow agent on the device, and a newly configured IP address overwrites the existing one.
3. Configure the sFlow collector information.	**sflow collector** *collector-id* { **ip** *ip-address* | **ipv6** *ipv6-address* } [**port** *port-number* | **datagram-size** *size* | **time-out** *seconds* | **description** *text*] *	By default, no sFlow collector information is configured.
4. (Optional.) Specify the source IP address of sFlow packets.	**sflow source** { **ip** *ip-address* | **ipv6** *ipv6-address* } *	By default, the source IP address is determined by routing.

Configuring flow sampling

Perform this task to configure flow sampling on an Ethernet interface. The sFlow agent performs the following tasks:

1. Samples packets on that interface according to the configured parameters.
2. Encapsulates the packets into sFlow packets.
3. Encapsulates the sFlow packets in the UDP packets and sends the UDP packets to the specified sFlow collector.

To configure flow sampling:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A

Step	Command	Remarks
3. (Optional.) Set the flow sampling mode.	**sflow sampling-mode** { **determine** | **random** }	The default setting is random. The **determine** keyword is not supported in the current software version. It is reserved for future support.
4. Enable flow sampling and specify the number of packets out of which flow sampling samples a packet on the interface.	**sflow sampling-rate** *rate*	By default, no flow sampling rate is configured.
5. (Optional.) Set the maximum number of bytes (starting from the packet header) that flow sampling can copy per packet.	**sflow flow max-header** *length*	The default setting is 128 bytes. HP recommends the default.
6. Specify the sFlow collector for flow sampling.	**sflow flow collector** *collector-id*	By default, no sFlow collector is specified for flow sampling.

Configuring counter sampling

Perform this task to configure counter sampling on an Ethernet interface. The sFlow agent performs the following tasks:

1. Periodically collects the counter information on that interface.
2. Encapsulates the counter information into sFlow packets.
3. Encapsulates the sFlow packets in the UDP packets and sends the UDP packets to the specified sFlow collector.

To configure counter sampling:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter Layer 2 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3. Enable counter sampling and set the counter sampling interval.	**sflow counter interval** *interval-time*	By default, counter sampling is disabled.
4. Specify the sFlow collector for counter sampling.	**sflow counter collector** *collector-id*	By default, no sFlow collector is specified for counter sampling.

Displaying and maintaining sFlow

Execute **display** commands in any view.

Task	Command
Display sFlow configuration.	**display sflow**

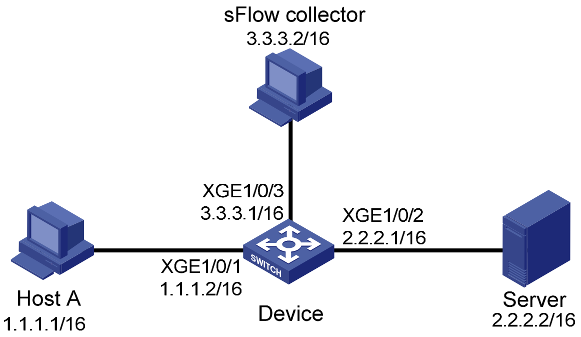
sFlow configuration example

Network requirements

As shown in Figure 63, perform the following tasks:

- Configure flow sampling in random mode and counter sampling on Ten-GigabitEthernet 1/0/1 of the device to monitor traffic on the port.
- Configure the device to send sampled information in sFlow packets through Ten-GigabitEthernet 1/0/3 to the sFlow collector.

Figure 63 Network diagram



Configuration procedure

1. Configure the IP addresses and subnet masks for interfaces, as shown in Figure 63. (Details not shown.)
2. Configure the sFlow agent and configure information about the sFlow collector:

   # Configure the IP address for the sFlow agent.

   ```
   <Device> system-view
   [Device] sflow agent ip 3.3.3.1
   ```

   # Configure information about the sFlow collector: specify the sFlow collector ID as 1, IP address as 3.3.3.2, port number as 6343 (default), and description as **netserver**.

   ```
   [Device] sflow collector 1 ip 3.3.3.2 description netserver
   ```
3. Configure counter sampling:

   # Enable counter sampling and set the counter sampling interval to 120 seconds on Ten-GigabitEthernet 1/0/1.

   ```
   [Device] interface ten-gigabitethernet 1/0/1
   [Device-Ten-GigabitEthernet1/0/1] sflow counter interval 120
   ```

   # Specify sFlow collector 1 for counter sampling.

   ```
   [Device-Ten-GigabitEthernet1/0/1] sflow counter collector 1
   ```
4. Configure flow sampling:

# Enable flow sampling and set the flow sampling mode to random and sampling interval to 4000.

```
[Device-Ten-GigabitEthernet1/0/1] sflow sampling-mode random
[Device-Ten-GigabitEthernet1/0/1] sflow sampling-rate 4000
```

# Specify sFlow collector 1 for flow sampling.

```
[Device-Ten-GigabitEthernet1/0/1] sflow flow collector 1
```

Verifying the configuration

# Verify that Ten-GigabitEthernet 1/0/1 enabled with sFlow is active, and sFlow is operating correctly.

```
[Device-Ten-GigabitEthernet1/0/1] display sflow
sFlow datagram version: 5
Global information:
Agent IP: 3.3.3.1(CLI)
Source address:
Collector information:
ID    IP              Port  Aging      Size VPN-instance Description
1     3.3.3.2         6343  N/A        1400              netserver
Port information:
Interface      CID   Interval(s) FID   MaxHLen Rate     Mode      Status
XGE1/0/1       1     120         1     128     4000     Random    Active
```

Troubleshooting sFlow configuration

The remote sFlow collector cannot receive sFlow packets

Symptom

The remote sFlow collector cannot receive sFlow packets.

Analysis

The possible reasons include:

- The sFlow collector is not specified.
- sFlow is not configured on the interface.
- The IP address of the sFlow collector specified on the sFlow agent is different from that of the remote sFlow collector.
- No IP address is configured for the Layer 3 interface that sends sFlow packets,
- An IP address is configured for the Layer 3 interface that sends sFlow packets. However, the UDP datagrams with this source IP address cannot reach the sFlow collector.
- The physical link between the device and the sFlow collector fails.
- The length of an sFlow packet is less than the sum of the following two values:
  - The length of the sFlow packet header.
  - The number of bytes that flow sampling can copy per packet.

Solution

To resolve the problem:

1. Verify that sFlow is correctly configured by using the **display sflow** command.
2. Verify that a correct IP address is configured for the device to communicate with the sFlow collector.
3. Verify that the physical link between the device and the sFlow collector is up.
4. Verify that the length of an sFlow packet is greater than the sum of the following two values:
   - The length of the sFlow packet header.
   - The number of bytes (HP recommends the default) that flow sampling can copy per packet.

Monitoring and maintaining processes

HP Comware V7 is a full-featured, modular, and scalable network operating system based on the Linux kernel. Comware V7 software features run the following types of independent processes:

- **User process**—Runs in user space. Most Comware V7 software features run user processes. Each process runs in an independent space so the failure of a process does not affect other processes. The system automatically monitors user processes. Comware V7 supports preemptive multithreading. A process can run multiple threads to support multiple activities. Whether a process supports multithreading depends on the software implementation.
- **Kernel thread**—Runs in kernel space. A kernel thread executes kernel code. It has a higher security level than a user process. If a kernel thread fails, the system breaks down. You can monitor the running status of kernel threads.

Displaying and maintaining processes

Commands described in this section apply to both user processes and kernel threads. You can execute these commands in any view.

The system identifies a process that consumes excessive memory or CPU resources as an anomaly source.

To display and maintain processes:

Task	Command
Display memory usage.	**display memory** [**slot** *slot-number* [**cpu** *cpu-number*]]
Display process state information.	**display process** [**all** | **job** *job-id* | **name** *process-name*] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display CPU usage for all processes.	**display process cpu** [**slot** *slot-number* [**cpu** *cpu-number*]]
Monitor process running state.	**monitor process** [**dumbtty**] [**iteration** *number*] [**slot** *slot-number* [**cpu** *cpu-number*]]
Monitor thread running state.	**monitor thread** [**dumbtty**] [**iteration** *number*] [**slot** *slot-number* [**cpu** *cpu-number*]]

For detailed information about the **display memory** [**slot** *slot-number*] command, see *Fundamentals Command Reference*.

Displaying and maintaining user processes

Execute **display** commands in any view and other commands in user view.

Task	Command
Display log information for all user processes.	**display process log** [**slot** *slot-number* [**cpu** *cpu-number*]]

Task	Command
Display memory usage for all user processes.	**display process memory** [**slot** *slot-number* [**cpu** *cpu-number*]]
Display heap memory usage for a user process.	**display process memory heap job** job-id [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display the addresses of memory blocks with a specified size used by a user process.	**display process memory heap job** *job-id* **size** *memory-size* [**offset** *offset-size*] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display memory content starting from a specified memory block for a user process.	**display process memory heap job** *job-id* **address** *starting-address* **length** *memory-length* [**slot** *slot-number* [**cpu** *cpu-number*]]
Display context information for process exceptions.	**display exception context** [**count** *value*] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display the core file directory.	**display exception filepath** [**slot** *slot-number* [**cpu** *cpu-number*]]
Enable or disable a process to generate core files for exceptions and set the maximum number of core files (which defaults to 1).	**process core** { **maxcore** *value* | **off** } { **job** *job-id* | **name** *process-name* } [**slot** *slot-number* [**cpu** *cpu-number*]]
Specify the directory for saving core files (the default directory is flash:/ on the master device).	**exception filepath** *directory*
Clear context information for process exceptions.	**reset exception context** [**slot** *slot-number* [**cpu** *cpu-number*]]

Monitoring kernel threads

Tasks in this section help you quickly identify thread deadloop and starvation problems and their causes.

Configuring kernel thread deadloop detection

△ CAUTION:

Inappropriate configuration of kernel thread deadloop detection can cause service problems or system breakdown. Make sure you understand the impact of this configuration on your network before you configure kernel thread deadloop detection.

Kernel threads share resources. If a kernel thread monopolizes the CPU, other threads cannot run, resulting in a deadloop.

This feature enables the device to detect deadloops. If a thread occupies the CPU for a specific interval, the device considers that a deadloop has occurred. It generates a deadloop message and reboots to remove the deadloop.

To configure kernel thread deadloop detection:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step		Command	Remarks
2.	Enable kernel thread deadloop detection.	**monitor kernel deadloop enable** [**slot** *slot-number* [**cpu** *cpu-number*]]	By default, this function is disabled.
3.	(Optional.) Set the interval for identifying a kernel thread deadloop.	**monitor kernel deadloop time** *interval* [**slot** *slot-number* [**cpu** *cpu-number*]]	The default is 8 seconds.
4.	(Optional.) Disable kernel thread deadloop detection for a kernel thread.	**monitor kernel deadloop exclude-thread** *tid* [**slot** *slot-number* [**cpu** *cpu-number*]]	After enabled, kernel thread deadloop detection monitors all kernel threads by default.

Configuring kernel thread starvation detection

△ CAUTION:

Inappropriate configuration of kernel thread starvation detection can cause service problems or system breakdown. Make sure you understand the impact of this configuration on your network before you configure kernel thread starvation detection.

Starvation occurs when a thread is unable to access shared resources.

Kernel thread starvation detection enables the system to detect and report thread starvation. If a thread is not executed within a specific interval, the system considers that a starvation has occurred, and generates a starvation message.

Thread starvation does not impact system operation. A starved thread can automatically run when certain conditions are met.

To configure kernel thread starvation detection:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enable kernel thread starvation detection.	**monitor kernel starvation enable** [**slot** *slot-number* [**cpu** *cpu-number*]]	By default, the function is disabled.
3.	(Optional.) Set the interval for identifying a kernel thread starvation.	**monitor kernel starvation time** *interval* [**slot** *slot-number* [**cpu** *cpu-number*]]	The default is 120 seconds.
4.	(Optional.) Disable kernel thread starvation detection for a kernel thread.	**monitor kernel starvation exclude-thread** *tid* [**slot** *slot-number* [**cpu** *cpu-number*]]	After enabled, kernel thread starvation detection monitors all kernel threads by default.

Displaying and maintaining kernel threads

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display kernel thread deadloop information.	**display kernel deadloop** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]

Task	Command
Display kernel thread deadloop detection configuration.	**display kernel deadloop configuration** [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread exception information.	**display kernel exception** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread reboot information.	**display kernel reboot** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread starvation information.	**display kernel starvation** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread starvation detection configuration.	**display kernel starvation configuration** [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear kernel thread deadloop information.	**reset kernel deadloop** [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear kernel thread exception information.	**reset kernel exception** [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear kernel thread reboot information.	**reset kernel reboot** [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear kernel thread starvation information.	**reset kernel starvation** [**slot** *slot-number* [**cpu** *cpu-number*]]

Configuring EAA

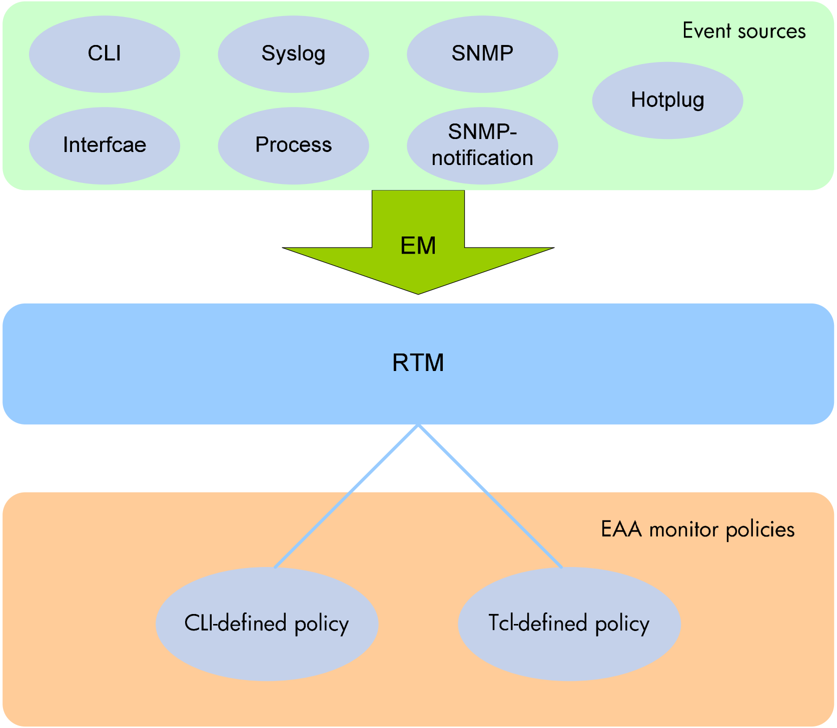
Overview

Embedded Automation Architecture (EAA) is a monitoring framework that enables you to self-define monitored events and actions to take in response to an event. It allows you to create monitor policies by using the CLI or Tcl scripts.

EAA framework

EAA framework includes a set of event sources, a set of event monitors, a real-time event manager (RTM), and a set of user-defined monitor policies, as shown in Figure 64.

Figure 64 EAA framework



Event sources

Event sources are software or hardware modules that produce events (see Figure 64). For example, the CLI module produces an event when you enter a command. The Syslog module (the information center) produces an event when it receives a log message.

Event monitors

EAA creates one event monitor for a monitor policy to monitor the system for the event specified in each monitor policy. An event monitor notifies the RTM to run the monitor policy when the monitored event occurs.

RTM

RTM manages the creation, state machine, and execution of monitor policies.

EAA monitor policies

A monitor policy specifies the event to monitor and actions to take when the event occurs.

You can configure EAA monitor policies by using the CLI or Tcl.

A monitor policy contains the following elements:

- One event.
- A minimum of one action.
- A minimum of one user role.
- One running time setting.

For more information, see "Elements in a monitor policy."

Elements in a monitor policy

Event

Table 19 shows types of events that EAA can monitor.

Table 19 Monitored events

Event type	Description
CLI	CLI event occurs in response to monitored operations performed at the CLI. For example, a command is entered, a question mark (?) is entered, or the **Tab** key is pressed to complete a command.
Syslog	Syslog event occurs when the information center receives the monitored log within a specific period. NOTE: The log that is generated by the EAA RTM does not trigger the monitor policy to run.
Process	Process event occurs in response to a state change (caused by an automatic system task) of the monitored process (such as an exception, shutdown, start, or restart).
Hotplug	Hotplug event occurs when the following situations occur: • Master/subordinate switchover occurs. • Member device is added to or removed from the IRF fabric.
Interface	Each interface event is associated with two user-defined thresholds: start and restart. An interface event occurs when the monitored interface traffic statistic crosses the start threshold in the following situations: • The statistic crosses the start threshold for the first time. • The statistic crosses the start threshold each time after it crosses the restart threshold.
SNMP	Each SNMP event is associated with two user-defined thresholds: start and restart. SNMP event occurs when the monitored MIB variable's value crosses the start threshold in the following situations: • The monitored variable's value crosses the start threshold for the first time. • The monitored variable's value crosses the start threshold each time after it crosses the restart threshold.

Event type	Description
SNMP-Notification	SNMP-Notification event occurs when the monitored MIB variable's value in an SNMP notification matches the specified condition. For example, the broadcast traffic rate on an Ethernet interface is equal to or greater than 30%.

Action

You can create a series of order-dependent actions to take in response to the event specified in the monitor policy.

The following are available actions:

- Executing a command.
- Sending a log.
- Enabling an active/standby switchover.
- Executing a reboot without saving the running configuration.

User role

For EAA to execute an action in a monitor policy, you must assign the policy the user role that has access to the action-specific commands and resources. If EAA lacks access to an action-specific command or resource, EAA does not perform the action and all the subsequent actions.

For example, a monitor policy has four actions numbered from 1 to 4. The policy has user roles that are required for performing actions 1, 3, and 4, but it does not have the user role required for performing action 2. When the policy is triggered, EAA executes only action 1.

For more information about user roles, see RBAC in *Fundamentals Configuration Guide*.

Runtime

Policy runtime limits the amount of time that the monitor policy can run from the time it is triggered. This setting prevents system resources from being occupied by incorrectly defined policies.

EAA environment variables

EAA environment variables decouple the configuration of action arguments from the monitor policy so you can modify a policy easily.

An EAA environment variable is defined as a <*variable_name variable_value*> pair and can be used in different policies. When you define an action, you can enter a variable name with a leading dollar sign (**$***variable_name*) instead of entering a value for an argument. EAA will replace the variable name with the variable value when it performs the action.

To change the value for an action argument, modify the value specified in the variable pair instead of editing each affected monitor policy.

EAA environment variables include system-defined variables and user-defined variables.

System-defined variables

System-defined variables are provided by default, and they cannot be created, deleted, or modified by users. System-defined variable names start with an underscore (_) sign, and variable values are set automatically by the system depending on the event setting in the policy that references the variables.

System-defined variables include the following types:

- **Public variable**—Available for any events.

- **Event-specific variable**—Available only for a type of event.

Table 20 shows all system-defined variables.

Table 20 System-defined EAA environment variables by event type

Variable name	Description
Any event:	
_event_id	Event ID.
_event_type	Event type.
_event_type_string	Event type description.
_event_time	Time when the event occurs.
_event_severity	Severity level of an event.
CLI:	
_cmd	Commands that are matched.
Syslog:	
_syslog_pattern	Log message content.
Hotplug:	
_slot	ID of the IRF member device where a hot swap event occurs.
Interface:	
_ifname	Interface name.
SNMP:	
_oid	OID of the MIB variable where an SNMP operation is performed.
_oid_value	Value of the MIB variable.
SNMP-Notification:	
_oid	OID that is included in the SNMP notification.
Process:	
_process_name	Process name.

User-defined variables

You can use user-defined variables for all types of events.

User-defined variable names can contain digits, characters, and the underscore sign (_), except that their leading character cannot be the underscore sign.

Configuring a user-defined EAA environment variable

Configure a user-defined EAA environment variable before you use it in an action.

To configure a user-defined EAA environment variable:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure a user-defined EAA environment variable.	**rtm environment** *env-name env-value*	By default, no user-defined environment variables are configured. The system provides the system-defined variables in Table 20.

Configuring a monitor policy

You can configure a monitor policy by using the CLI or Tcl.

Configuration restrictions and guidelines

When you configure monitor policies, follow these restrictions and guidelines:

- Make sure the actions in different policies do not conflict. Policy execution result will be unpredictable if policies that conflict in actions are running concurrently.
- You can assign the same policy name to a CLI-defined policy and a Tcl-defined policy, but you cannot assign the same name to policies that are the same type.
- The system executes the actions in a policy in ascending order of action IDs. When you add actions to a policy, you must make sure the execution order is correct.

Configuring a monitor policy from the CLI

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter CLI-defined policy view.	**rtm cli-policy** *policy-name*	If the policy does not exist, this command creates the policy first.

Step	Command	Remarks
3. Configure an event in the policy.	• Configure a CLI event: **event cli** { **async** [**skip**] | **sync** } **mode** { **execute** | **help** | **tab** } **pattern** *regular-exp* • Configure a hotplug event: **event hotplug** [**slot** *slot-number*] • Configure an interface event: **event interface** *interface-type interface-number* **monitor-obj** *monitor-obj* **start-op** *start-op* **start-val** *start-val* **restart-op** *restart-op* **restart-val** *restart-val* [**interval** *interval*] • Configure a process event: **event process** { **exception** | **restart** | **shutdown** | **start** } [**name** *process-name* [**instance** *instance-id*]] [**slot** *slot-number*] • Configure an SNMP event: **event snmp oid** *oid* **monitor-obj** { **get** | **next** } **start-op** *start-op* **start-val** *start-val* **restart-op** *restart-op* **restart-val** *restart-val* [**interval** *interval*] • Configure an SNMP-Notification event: **event snmp-notification oid** *oid* **oid-val** *oid-val* **op** *op* [**drop**] • Configure a Syslog event: **event syslog priority** *level* **msg** *msg* **occurs** *times* **period** *period*	By default, a monitor policy does not contain an event. You can configure only one event in a monitor policy. If the monitor policy already contains an event, the new event overrides the old event.
4. Configure the actions to take when the event occurs.	• Configure the action to execute a command: **action** *number* **cli** *command-line* • Configure a reboot action: **action** *number* **reboot** [**slot** *slot-number*] • Configure a logging action: **action** *number* **syslog priority** *level* **facility** *local-number* **msg** *msg* • Configure an active/standby switchover action: **action** *number* **switchover**	By default, a monitor policy does not contain any actions. Repeat this step to add a maximum of 232 actions to the policy. You can reference a variable name in the **$***variable_name* format instead of specifying a value for an argument when you define an action.

Step	Command	Remarks
5. (Optional.) Assign a user role to the policy.	**user-role** *role-name*	By default, a monitor policy contains user roles that its creator had at the time of policy creation. A monitor policy supports a maximum of 64 valid user roles. User roles added after this limit is reached do not take effect. An EAA policy cannot have both the **security-audit** user role and any other user roles. Any previously assigned user roles are automatically removed when you assign the **security-audit** user role to the policy. The previously assigned **security-audit** user role is automatically removed when you assign any other user roles to the policy.
6. (Optional.) Configure the policy runtime.	**running-time** *time*	The default runtime is 20 seconds.
7. Enable the policy.	**commit**	By default, CLI-defined policies are not enabled. A CLI-defined policy can take effect only after you perform this step.

Configuring a monitor policy by using Tcl

Step	Command	Remarks
1. Edit a Tcl script file (see Table 21).	N/A	The supported Tcl version is 8.5.8.
2. Download the file to the device by using FTP or TFTP.	N/A	For more information about using FTP and TFTP, see *Fundamentals Configuration Guide*.
3. Enter system view.	**system-view**	N/A
4. Create a Tcl-defined policy and bind it to the Tcl script file.	**rtm tcl-policy** *policy-name* *tcl-filename*	By default, the system does not have Tcl policies. This step enables the Tcl-defined policy. To revise the Tcl script of a policy, you must suspend all monitor policies first, and then resume the policies after you finish revising the script. The system cannot execute a Tcl-defined policy if you edit its Tcl script without suspending policies.

Write a Tcl script in two lines for a monitor policy, as shown in Table 21.

Table 21 Tcl script requirements

Line	Content	Requirements
Line 1	Event, user roles, and policy runtime	This line must take the following format: **::comware::rtm::event_register** *eventname arg1 arg2 arg3* …**user-role** *rolename1* | [**user-role** *rolename2* | [ ]][**running-time** *running-time*]
Line 2	Actions	You can reference a variable name in the **$***variable_name* format instead of specifying a value for an argument when you define an action. The following actions are available: • Standard Tcl commands. • EAA-specific Tcl commands. • Commands supported by the device.

Suspending monitor policies

This task suspends all CLI-defined and Tcl-defined monitor policies except for the policies that are running.

To suspend monitor policies:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Suspend monitor policies.	**rtm scheduler suspend**	To resume monitor polices, use the **undo rtm scheduler suspend** command.

Displaying and maintaining EAA settings

Execute **display** commands in any view.

Task	Command
Display user-defined EAA environment variables.	**display rtm environment** [*var-name*]
Display EAA monitor policies.	**display rtm policy** { **active** | **registered** } [*policy-name*]

Configuration examples

CLI-defined policy configuration example

Network requirements

Configure a policy from the CLI to monitor the event that occurs when a question mark (?) is entered at the command line that contains letters and digits.

When the event occurs, the system executes the command and sends the log message "hello world" to the information center.

Configuration procedure

# Create the CLI-defined policy **test** and enter its view.

```
<Sysname> system-view
[Sysname] rtm cli-policy test
```

# Add a CLI event that occurs when a question mark (?) is entered at any command line that contains letters and digits.

```
[Sysname-rtm-test] event cli async mode help pattern [a-zA-Z0-9]
```

# Add an action that sends the message "hello world" with priority **4** from the logging facility **local3** when the event occurs.

```
[Sysname-rtm-test] action 0 syslog priority 4 facility local3 msg “hello world”
```

# Add an action that enters system view when the event occurs.

```
[Sysname-rtm-test] action 2 cli system-view
```

# Set the policy runtime to 2000 seconds. The system stops executing the policy and displays an execution failure message if it fails to complete policy execution within 2000 seconds.

```
[Sysname-rtm-test] running-time 2000
```

# Specify the **network-admin** user role for executing the policy.

```
[Sysname-rtm-test] user-role network-admin
```

# Enable the policy.

```
[Sysname-rtm-test] commit
```

Verifying the configuration

# Display information about the policy.

```
[Sysname-rtm-test] display rtm policy registered
Total number: 1
Type   Event      TimeRegistered           PolicyName
CLI    CLI        May 07 02:08:17 2013     test
```

# Enable the information center to output log messages to the current monitoring terminal.

```
[Sysname-rtm-test] return
<Sysname> terminal monitor
```

# Enter a question mark (?) at a command line that contains a letter **d**. Verify that the system displays the "hello world" message and a policy successfully executed message on the terminal screen.

```
<Sysname> d?
  debugging
  delete
  diagnostic-logfile
  dir
  display
```

```
<Sysname>d%May  7 02:10:03:218 2013 Sysname RTM/4/RTM_ACTION: "hello world"
%May  7 02:10:04:176 2013 Sysname RTM/6/RTM_POLICY: CLI policy test is running
successfully.
```

CLI-defined policy with EAA environment variables configuration example

Network requirements

Define an environment variable to match the IP address 1.1.1.1.

Configure a policy from the CLI to monitor the event that occurs when a command line that contains **loopback0** is executed. In the policy, use the environment variable for IP address assignment.

When the event occurs, the system performs the following tasks:

- Creates the Loopback 0 interface.
- Assigns 1.1.1.1/24 to the interface.
- Sends the matching command line to the information center.

Configuration procedure

# Configure an EAA environment variable for IP address assignment. The variable name is **loopback0IP**, and the variable value is **1.1.1.1**.

```
<Sysname> system-view
[Sysname] rtm environment loopback0IP 1.1.1.1
```

# Create the CLI-defined policy **test** and enter its view.

```
[Sysname] rtm cli-policy test
```

# Add a CLI event that occurs when a command line that contains **loopback0** is executed.

```
[Sysname-rtm-test] event cli async mode execute pattern loopback0
```

# Add an action that enters system view when the event occurs.

```
[Sysname-rtm-test] action 0 cli system-view
```

# Add an action that creates the interface Loopback 0 and enters loopback interface view.

```
[Sysname-rtm-test] action 1 cli interface loopback 0
```

# Add an action that assigns the IP address 1.1.1.1 to Loopback 0. The **loopback0IP** variable is used in the action for IP address assignment.

```
[Sysname-rtm-test] action 2 cli ip address $loopback0IP 24
```

# Add an action that sends the matching **loopback0** command with a priority of 0 from the logging facility **local7** when the event occurs.

```
[Sysname-rtm-test] action 3 syslog priority 0 facility local7 msg $_cmd
```

# Specify the **network-admin** user role for executing the policy.

```
[Sysname-rtm-test] user-role network-admin
```

# Enable the policy.

```
[Sysname-rtm-test] commit
```

```
[Sysname-rtm-test] return
<Sysname>
```

Verifying the configuration

# Enable the information center to output log messages to the current monitoring terminal.

```
<Sysname> terminal monitor
```

# Execute the **loopback0** command. Verify that the system displays the **loopback0** message and a policy successfully executed message on the terminal screen.

```
<Sysname> loopback0
<Sysname>
%Jan  3 09:46:10:592 2014 Sysname RTM/0/RTM_ACTION: loopback0
%Jan  3 09:46:10:613 2014 Sysname RTM/6/RTM_POLICY: CLI policy test is running
successfully.
```

# Verify that Loopback 0 has been created and assigned the IP address 1.1.1.1.

```
<Sysname> terminal monitor
<Sysname> display interface loopback brief
Brief information on interface(s) under route mode:
Link: ADM - administratively down; Stby - standby
Protocol: (s) - spoofing
Interface            Link Protocol Main IP         Description
Loop0                UP   UP(s)    1.1.1.1
<Sysname>
```

Tcl-defined policy configuration example

Network requirements

Use Tcl to create a monitor policy on the device. This policy must meet the following requirements:

- EAA sends the log message "rtm_tcl_test is running" when a command that contains the **display this** string is entered.
- The system executes the command only after it executes the policy successfully.

Configuration procedure

# Edit a Tcl script file (rtm_tcl_test.tcl, in this example) for EAA to send the message "rtm_tcl_test is running" when a command that contains the **display this** string is executed.

```
::comware::rtm::event_register cli sync mode execute pattern display this user-role
network-admin
::comware::rtm::action syslog priority 1 facility local4 msg rtm_tcl_test is running
```

# Download the Tcl script file from the TFTP server at **1.2.1.1**.

```
<Sysname> tftp 1.2.1.1 get rtm_tcl_test.tcl
```

# Create the Tcl-defined policy **test** and bind it to the Tcl script file.

```
<Sysname> system-view
[Sysname] rtm tcl-policy test rtm_tcl_test.tcl
[Sysname] quit
```

Verifying the configuration

# Display information about the policy.

```
<Sysname> display rtm policy registered
```

```
Total number: 1
Type  Event      TimeRegistered       PolicyName
TCL   TCL        Apr 21 16:33:00 2012 test
```

# Enable the information center to output log messages to the current monitoring terminal.

```
<Sysname> terminal monitor
```

# Execute the **display this** command. Verify that the system displays the "rtm_tcl_test is running" message and a message that the policy is being successfully executed.

```
<Sysname> display this
#
return
<Sysname>%Jun  4 15:02:30:354 2013 Sysname RTM/1/RTM_ACTION: rtm_tcl_test is running
%Jun  4 15:02:30:382 2013 Sysname RTM/6/RTM_POLICY: TCL policy test is running
successfully.
```

Configuring CWMP

Overview

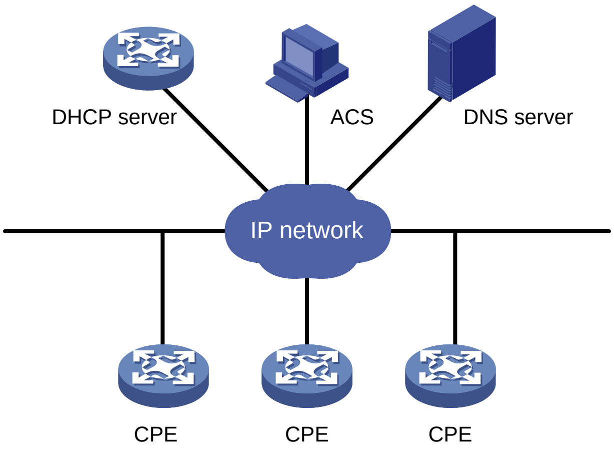
CPE WAN Management Protocol (CWMP), also called "TR-069," is a DSL Forum technical specification for remote management of home network devices.

The protocol was initially designed to provide remote autoconfiguration through a server for large numbers of dispersed end-user devices in DSL networks. However, it has been increasingly used on other types of networks, including Ethernet, for remote autoconfiguration.

CWMP network framework

Figure 1 shows a basic CWMP network framework.

Figure 1 CWMP network framework



A basic CWMP network includes the following network elements:

- **ACS**—Autoconfiguration server, the management device in the network.
- **CPE**—Customer premises equipment, the managed device in the network.
- **DNS server**—Domain name system server. CWMP defines that the ACS and the CPE use URLs to identify and access each other. DNS is used to resolve the URLs.
- **DHCP server**—Assigns ACS attributes along with IP addresses to CPEs when the CPEs are powered on. DHCP server is optional in CWMP. With a DHCP server, you do not need to configure ACS attributes manually on each CPE. The CPEs contact the ACS automatically when they are powered on for the first time.

The device is operating as a CPE in the CWMP framework.

Basic CWMP functions

The ACS identifies different categories of CPEs by provision code. You can use the ACS to autoconfigure and upgrade each category of CPEs in bulk.

Autoconfiguration

You can create configuration files for different categories of CPEs on the ACS. The ACS identifies the configuration file for a CPE by its provision code.

The following are methods available for the ACS to issue configuration to the CPE:

- Transfers the configuration file to the CPE, and specifies the file as the next-startup configuration file. At a reboot, the CPE starts up with the ACS-specified configuration file.
- Runs the configuration in the CPE's RAM. The configuration takes effect immediately on the CPE. For the running configuration to survive a reboot, you must save the configuration on the CPE.

Software image management

The ACS can manage CPE software upgrade.

When the ACS finds a software version update, the ACS notifies the CPE to download the software image file from a specific location. The location can be the URL of the ACS or an independent file server.

The CPE notifies the ACS of the download result (success or failure) when it completes a download attempt. The CPE downloads the specified image file only when the file passes validity verification.

Data backup

The ACS can require the CPE to upload a configuration or log file to a specific location. The destination location can be the ACS or a file server.

Status and performance monitoring

The CPE allows the ACS to monitor the status and performance objects in Table 22.

Table 22 CPE status and performance objects available for the ACS to monitor

Category	Objects
Device information	Manufacturer ManufacturerOUI SerialNumber HardwareVersion SoftwareVersion
Operating status and information	DeviceStatus UpTime
Configuration file	ConfigFile

Category	Objects
CWMP settings	ACS URL ACS username ACS password PeriodicInformEnable PeriodicInformInterval PeriodicInformTime ConnectionRequestURL (CPE URL) ConnectionRequestUsername (CPE username) ConnectionRequestPassword (CPE password)

How CWMP works

CWMP uses remote procedure call (RPC) methods for bidirectional communication between CPE and ACS. The RPC methods are encapsulated in HTTP or HTTPS.

RPC methods

Table 23 shows the primary RPC methods used in CWMP.

Table 23 RPC methods

RPC method	Description
Get	The ACS obtains the values of parameters on the CPE.
Set	The ACS modifies the values of parameters on the CPE.
Inform	The CPE sends an Inform message to the ACS for the following purposes: • Initiates a connection to the ACS. • Reports configuration changes to the ACS. • Periodically updates CPE settings to the ACS.
Download	The ACS requires the CPE to download a configuration or software image file from a specific URL for software or configuration update.
Upload	The ACS requires the CPE to upload a file to a specific URL.
Reboot	The ACS reboots the CPE remotely for the CPE to complete an upgrade or recover from an error condition.

Autoconnect between ACS and CPE

The CPE connects to the ACS automatically after it obtains the DNS server address and basic ACS parameters (ACS URL and authentication username and password). You can configure this information manually on the CPE, through a DHCP server, or through the ACS.

After establishing a connection, the ACS can issue configuration and software images to the CPE. If the connection is disconnected before a session is complete, the CPE retries the failed connection automatically. The retry attempt continues until the connection is established again or the specified retry limit is reached.

Depending on the configuration, the CPE can also connect to the ACS regularly or at a scheduled time to update its information with the ACS.

NOTE:

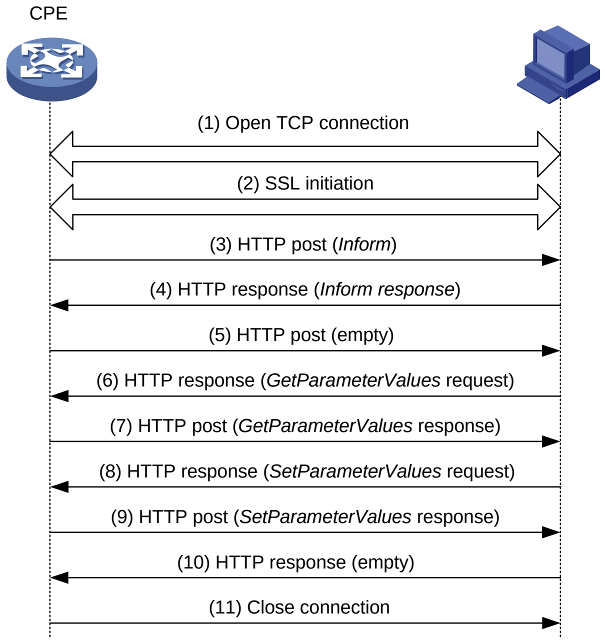
For the CPE to complete autoconfiguration at its initial startup, HP recommends that you use a DHCP server. The DHCP option for ACS parameter assignment is option 43. For more information about DHCP, see *Layer 3—IP Services Configuration Guide*.

CWMP connection establishment

As shown in Figure 2, the CPE and the ACS use the following process to establish a connection:

1. After obtaining the basic ACS parameters, the CPE initiates a TCP connection to the ACS.
2. If HTTPS is used, the CPE and the ACS initialize SSL for a secure HTTP connection.
3. The CPE sends an Inform message in HTTPS to initiate a CWMP session.
4. After the CPE passes authentication, the ACS returns an Inform response to establish the session.
5. After sending all requests, the CPE sends an empty HTTP post message.
6. If the ACS wants to point the CPE to a new ACS URL, the ACS queries the ACS URL set on the CPE.
7. The CPE replies with its ACS URL setting.
8. The ACS sends a Set request to modify the ACS URL on the CPE.
9. After the ACS URL is modified, the CPE sends a response.
10. The ACS sends an empty HTTP message to notify the CPE that it has no other requests.
11. The CPE closes the connection, and then initiates a new connection to the new ACS URL.

Figure 2 CWMP message interaction procedure


CPE
(1) Open TCP connection
(2) SSL initiation
(3) HTTP post (*Inform*)
(4) HTTP response (*Inform response*)
(5) HTTP post (empty)
(6) HTTP response (*GetParameterValues* request)
(7) HTTP post (*GetParameterValues* response)
(8) HTTP response (*SetParameterValues* request)
(9) HTTP post (*SetParameterValues* response)
(10) HTTP response (empty)
(11) Close connection

Configuration task list

To use CWMP, you must enable CWMP from the CLI. You can then configure ACS and CPE attributes from the CPE's CLI, the DHCP server, or the ACS.

For an attribute, the CLI- and ACS-assigned values have higher priority than the DHCP-assigned value. The CLI- and ACS-assigned values overwrite each other, whichever is assigned later.

This document only describes configuring ACS and CPE attributes from the CLI and DHCP server. For more information about configuring and using the ACS, see ACS documentation.

To configure CWMP, perform the following tasks:

Tasks at a glance	Remarks
(Required.) Enabling CWMP from the CLI	To use CWMP, you must enable CWMP from the CLI.
Configuring ACS attributes: • (Required.) Configuring the preferred ACS attributes ○ Assigning ACS attributes from the DHCP server ○ Configuring the preferred ACS attributes from the CLI • (Optional.) Configuring the default ACS attributes from the CLI	The preferred ACS attributes are configurable from the CPE's CLI, DHCP server, and ACS. The default ACS attributes are configurable only from the CLI.
(Optional.) Configuring CPE attributes: • Configuring ACS authentication parameters • Configuring the provision code • Configuring the CWMP connection interface • Configuring autoconnect parameters ○ Configuring the periodic Inform feature ○ Scheduling a connection initiation ○ Configuring the maximum number of connection retries ○ Configuring the close-wait timer • Enabling NAT traversal for the CPE • Specifying an SSL client policy for HTTPS connection to ACS	All CPE attributes are configurable from the CLI and ACS except for the following attributes: • CWMP connection interface • NAT traversal • Maximum number of connection retries • SSL client policy for HTTPS These attributes are configurable only from the CLI.

Enabling CWMP from the CLI

You must enable CWMP for other CWMP settings to take effect, whether they are configured from the CLI, or assigned through the DHCP server or ACS.

To enable CWMP:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter CWMP view.	**cwmp**	N/A
3. Enable CWMP.	**cwmp enable**	By default, CWMP is disabled.

Configuring ACS attributes

You can configure two sets of ACS attributes for the CPE: preferred and default.

- The preferred ACS attributes are configurable from the CPE's CLI, the DHCP server, and ACS. For an attribute, the CLI- and ACS-assigned values have higher priority than the DHCP-assigned value. The CLI- and ACS-assigned values overwrite each other.
- The default ACS attributes are configurable only from the CLI.

The CPE uses the default ACS attributes for connection establishment only when it is not assigned a preferred ACS URL from the CLI, ACS, or DHCP server.

Configuring the preferred ACS attributes

Assigning ACS attributes from the DHCP server

You can use DHCP option 43 to assign the ACS URL and ACS login authentication username and password.

If the DHCP server is an HP device, you can configure DHCP option 43 by using the **option 43 hex 01** *length URL username password* command.

- *length*—A hexadecimal number that indicates the total length of the *length*, *URL*, *username*, and *password* arguments, including the spaces between these arguments. No space is allowed between the **01** keyword and the length value.
- *URL*—ACS URL.
- *username*—Username for the CPE to authenticate to the ACS.
- *password*—Password for the CPE to authenticate to the ACS.

NOTE:

The ACS URL, username and password must use the hexadecimal format and be space separated.

The following example configures the ACS address as **http://169.254.76.31:7547/acs**, username as **1234**, and password as **5678**:

```
<Sysname> system-view
[Sysname] dhcp server ip-pool 0
[Sysname-dhcp-pool-0] option 43 hex
0127687474703A2F2F3136392E3235342E37362E33313A373534372F61637320313233342035363738
```

Table 24 Hexadecimal forms of the ACS attributes

Attribute	Attribute value	Hexadecimal form
Length	39 characters	27
ACS URL	http://169.254.76.31/acs	687474703A2F2F3136392E3235342E37362E33313A373534372F61637320 NOTE: The two ending digits (20) represent the space.
ACS connect username	1234	3132333420 NOTE: The two ending digits (20) represent the space.

Attribute	Attribute value	Hexadecimal form
ACS connect password	5678	35363738

For more information about DHCP and DHCP Option 43, see *layer 3—IP Services Configuration Guide*.

Configuring the preferred ACS attributes from the CLI

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter CWMP view.	**cwmp**	N/A
3. Configure the preferred ACS URL.	**cwmp acs url** *url*	By default, no preferred ACS URL has been configured.
4. Configure the username for authentication to the preferred ACS URL.	**cwmp acs username** *username*	By default, no username has been configured for authentication to the preferred ACS URL.
5. (Optional.) Configure the password for authentication to the preferred ACS URL.	**cwmp acs password** { **cipher** | **simple** } *password*	By default, no password has been configured for authentication to the preferred ACS URL.

Configuring the default ACS attributes from the CLI

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter CWMP view.	**cwmp**	N/A
3. Configure the default ACS URL.	**cwmp acs default url** *url*	By default, no default ACS URL has been configured.
4. Configure the username for authentication to the default ACS URL.	**cwmp acs default username** *username*	By default, no username has been configured for authentication to the default ACS URL.
5. (Optional.) Configure the password for authentication to the default ACS URL.	**cwmp acs default password** { **cipher** | **simple** } *password*	By default, no password has been configured for authentication to the default ACS URL.

Configuring CPE attributes

You can assign CPE attribute values to the CPE from the CPE's CLI or the ACS. The CLI- and ACS-assigned values overwrite each other, whichever is assigned later.

For more information about the configuration methods supported for each CPE attribute, see "Configuration task list."

Configuring ACS authentication parameters

To protect the CPE against unauthorized access, configure a CPE username and password for ACS authentication. When an ACS initiates a connection to the CPE, the ACS must provide the correct username and password.

NOTE:

The password setting is optional. You may choose to use only a username for authentication.

To configure ACS authentication parameters:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter CWMP view.	**cwmp**	N/A
3.	Configure the username for authentication to the CPE.	**cwmp cpe username** *username*	By default, no username has been configured for authentication to the CPE.
4.	(Optional.) Configure the password for authentication to the CPE.	**cwmp cpe password** { **cipher** | **simple** } *password*	By default, no password has been configured for authentication to the CPE.

Configuring the provision code

The ACS uses the provision code to identify services assigned to each CPE. For correct configuration deployment, make sure the same provision code is configured on the CPE and the ACS.

To configure the provision code:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter CWMP view.	**cwmp**	N/A
3.	Configure the provision code.	**cwmp cpe provision-code** *provision-code*	The default provision code is **PROVISIONINGCODE**.

Configuring the CWMP connection interface

The CWMP connection interface is the interface that the CPE uses to communicate with the ACS. To establish a CWMP connection, the CPE sends the IP address of this interface in the Inform messages, and the ACS replies to this IP address.

Typically, the CPE selects the CWMP connection interface automatically.

If the interface that connects the CPE to the ACS is the only Layer 3 interface that has an IP address on the device, you do not need to specify the CWMP connection interface.

If the CPE has multiple Layer 3 interfaces, specify the interface that connects to the ACS as the CWMP connection interface. This manual setting avoids the risk of incorrect CWMP connection interface selection in an automatic selection process.

To configure the CWMP connection interface:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter CWMP view.	**cwmp**	N/A
3.	Specify the interface that connects to the ACS as the CWMP connection interface.	**cwmp cpe connect interface** *interface-type interface-number*	No CWMP connection interface is specified.

Configuring autoconnect parameters

You can configure the CPE to connect to the ACS periodically, or at a schedule time for configuration or software update. To protect system resources, limit the number of retries that the CPE can make to connect to the ACS.

Configuring the periodic Inform feature

To connect to the ACS periodically for CPE information update:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter CWMP view.	**cwmp**	N/A
3.	Enable the periodic Inform feature.	**cwmp cpe inform interval enable**	By default, this function is disabled.
4.	(Optional.) Configure the Inform interval.	**cwmp cpe inform interval** *seconds*	By default, the CPE sends an Inform message to start a session every 600 seconds.

Scheduling a connection initiation

To connect to the ACS for configuration or software update at a scheduled time:

Step		Command	Remarks
1.	Enter system view.	**system-view**	N/A
2.	Enter CWMP view.	**cwmp**	N/A
3.	Schedule a connection initiation.	**cwmp cpe inform time** *time*	By default, no connection initiation has been scheduled.

Configuring the maximum number of connection retries

The CPE retries a connection automatically when one of the following events occurs:

- The CPE fails to connect to the ACS.
- The connection is disconnected before the session on the connection is completed.

The CPE considers a connection attempt as having failed when the close-wait timer expires. This timer starts when the CPE sends an Inform request. If the CPE fails to receive a response before the timer expires, the CPE resends the Inform request.

To configure the maximum number of connection retries that the CPE can make:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter CWMP view.	**cwmp**	N/A
3. Configure the maximum number of connection retries.	**cwmp cpe connect retry** *times*	By default, the CPE retries a failed connection until the connection is established.

Configuring the close-wait timer

The close-wait timer specifies the amount of time the connection to the ACS can be idle before it is terminated. The CPE terminates the connection to the ACS if no traffic is transmitted before the timer expires.

The timer also specifies the maximum amount of time the CPE waits for the response to a session request. The CPE determines that its session attempt has failed when the timer expires.

To configure the close-wait timer for the CPE:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter CWMP view.	**cwmp**	N/A
3. Set the close-wait timer.	**cwmp cpe wait timeout** *seconds*	By default, the close-wait timer is 30 seconds.

Enabling NAT traversal for the CPE

For the connection request initiated from the ACS to reach the CPE, you must enable NAT traversal feature on the CPE when a NAT gateway resides between the CPE and the ACS.

The NAT traversal feature complies with RFC 3489 Simple Traversal of UDP Through NATs (STUN). The feature enables the CPE to discover the NAT gateway, and obtain an open NAT binding (a public IP address and port binding) through which the ACS can send unsolicited packets. The CPE sends the binding to the ACS when it initiates a connection to the ACS. For the connection requests sent by the ACS at any time to reach the CPE, the CPE maintains the open NAT binding.

To enable NAT traversal on the CPE:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter CWMP view.	**cwmp**	N/A
3. Enable NAT traversal.	**cwmp cpe stun enable**	By default, NAT traversal is disabled on the CPE.

Specifying an SSL client policy for HTTPS connection to ACS

CWMP uses HTTP or HTTPS for data transmission. If the ACS uses HTTPS for secure access, its URL begins with **https://**. You must configure an SSL client policy for the CPE to authenticate the ACS for

HTTPS connection establishment. For more information about configuring SSL client policies, see *Security Configuration Guide*.

To specify an SSL client policy for the CPE to establish an HTTPS connection to the ACS:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter CWMP view.	**cwmp**	N/A
3. Specify an SSL client policy.	**ssl client-policy** *policy-name*	By default, no SSL client policy is specified.

Displaying and maintaining CWMP

Execute **display** commands in any view.

Task	Command
Display CWMP configuration.	**display cwmp configuration**
Display the current status of CWMP.	**display cwmp status**

CWMP configuration example

Network requirements

As shown in Figure 3, use HP IMC BIMS as the ACS to bulk-configure the devices (CPEs), and assign ACS attributes to the CPEs from the DHCP server.

The configuration files for the devices in equipment rooms A and B are **configure1.cfg** and **configure2.cfg**, respectively.

Figure 3 Network diagram

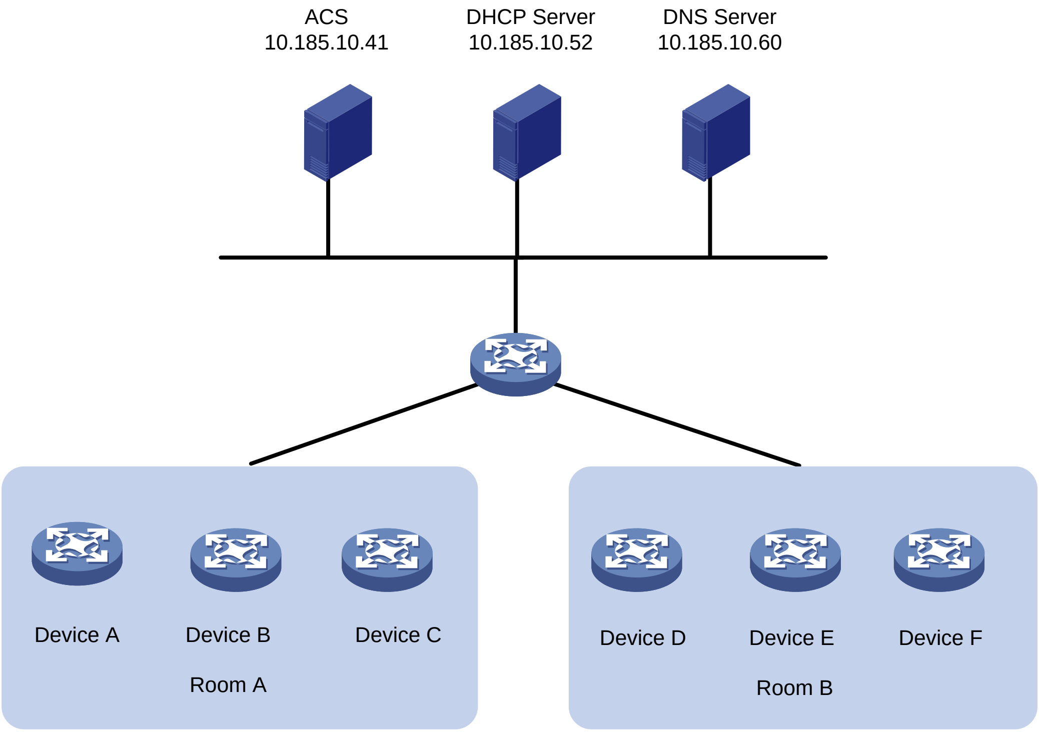


Table 25 shows the ACS attributes for the CPEs to connect to the ACS.

Table 25 ACS attributes

Item	Setting
Preferred ACS URL	http://10.185.10.41:8080/acs
ACS username	Admin
ACS password	12345

Table 26 lists serial numbers of the CPEs.

Table 26 CPE list

Room	Device	Serial number
A	Device A	210231A95YH10C000045
	Device B	210235AOLNH12000010
	Device C	210235AOLNH12000015
B	Device D	210235AOLNH12000017
	Device E	210235AOLNH12000020
	Device F	210235AOLNH12000022

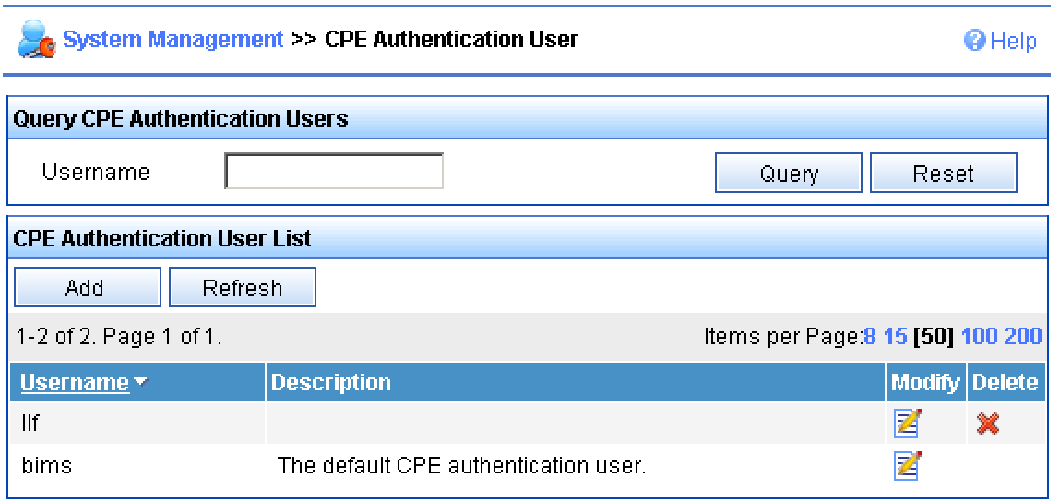
Configuration procedure

Configuring the ACS

1. Log in to the ACS:
   a. Launch a Web browser on the ACS configuration terminal.
   b. In the address bar of the Web browser, enter the ACS URL and port number. This example uses **http://10.185.10.41:8080/imc**.
   c. On the login page, enter the ACS login username and password, and then click **Login**.
2. Create a CPE user account:
   a. Select **Service** > **System Management** > **CPE Authentication User** from the top navigation bar.

      The CPE authentication user configuration page appears.

      Figure 4 CPE authentication user configuration page



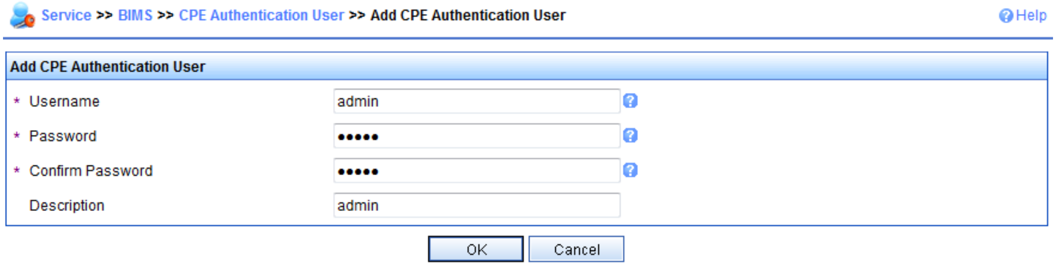
System Management >> CPE Authentication User
Help
Query CPE Authentication Users
Username
Query
Reset
CPE Authentication User List
Add
Refresh

1-2 of 2. Page 1 of 1. Items per Page:8 15 [50] 100 200

Username	Description	Modify	Delete
llf		📝	✗
bims	The default CPE authentication user.	📝	

   b. Click **Add**.
   c. Enter the username and password for authentication to the ACS, and then click **OK**.

      Figure 5 Adding a CPE user account



Service >> BIMS >> CPE Authentication User >> Add CPE Authentication User
Help
Add CPE Authentication User
OK
Cancel

* Username	admin
* Password	•••••
* Confirm Password	•••••
Description	admin

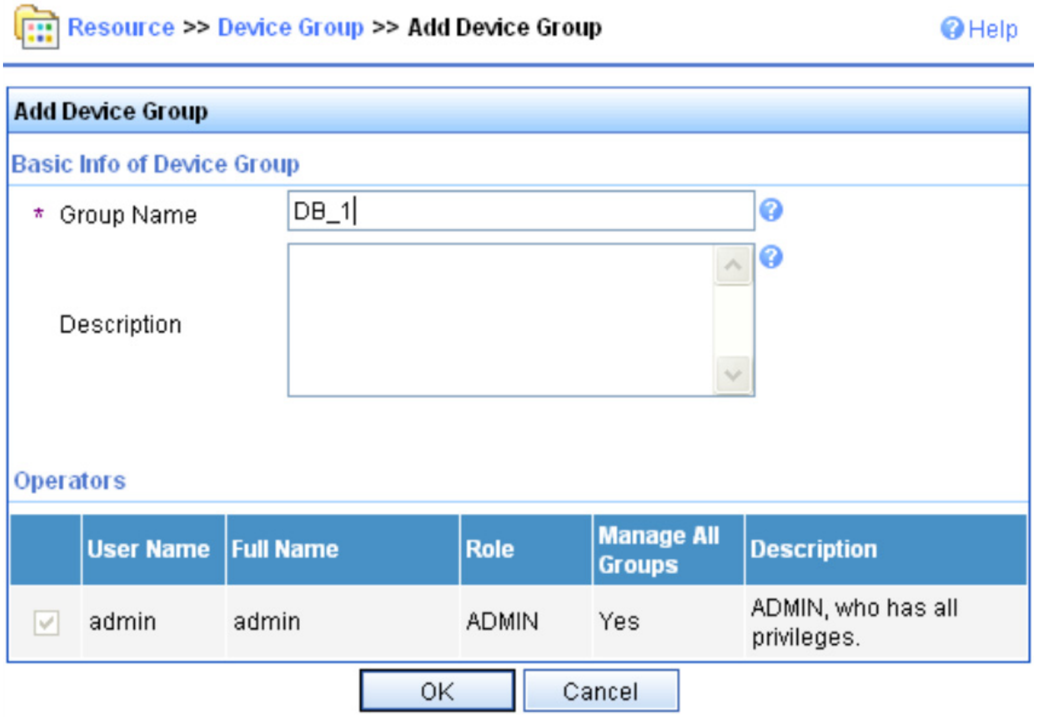
3. Add device groups and device classes for devices in equipment rooms A and B:

   This example assigns all devices to the same device group, and assigns the devices in two equipment rooms to different device classes.

   a. Select **Service** > **Resource** > **Device Group** from the top navigation bar.
   b. Click **Add**.

c. On the **Add Device Group** page, enter a service group name (for example, **DB_1**), and then click **OK**.

Figure 6 Adding a device group


Resource >> Device Group >> Add Device Group
Help
Add Device Group
Basic Info of Device Group
Operators
OK
Cancel

* Group Name	DB_1
Description	

	User Name	Full Name	Role	Manage All Groups	Description
☑	admin	admin	ADMIN	Yes	ADMIN, who has all privileges.

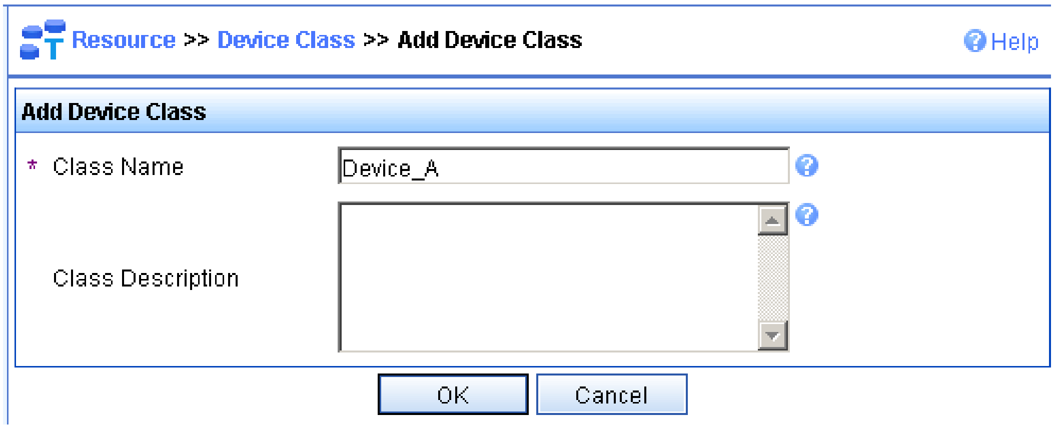
d. Select **Service** > **Resource** > **Device Class** from the top navigation bar.

e. Click **Add**.

f. On the **Add Device Class** page, enter a device class name for devices in equipment room A, and then click **OK**.

In this example, the device class for devices in equipment room A is **Device_A**.

Figure 7 Adding a device class



* Class Name	Device_A
Class Description	

g. Repeat the previous two steps to create a device class for devices in equipment room B.

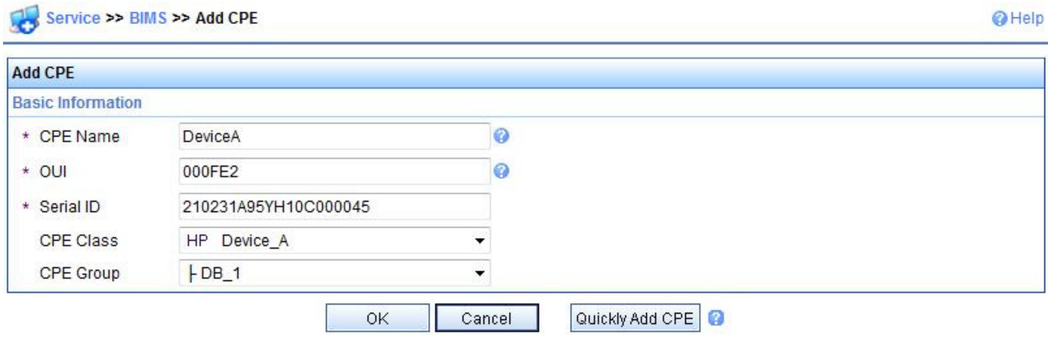
4. Add the devices as CPEs:

a. Select **Service** > **BIMS** > **Add CPE** from the top navigation bar.

b. On the **Add CPE** page, enter or select basic settings for device A, and then click **OK**.

c. Repeat the previous two steps to add other devices.

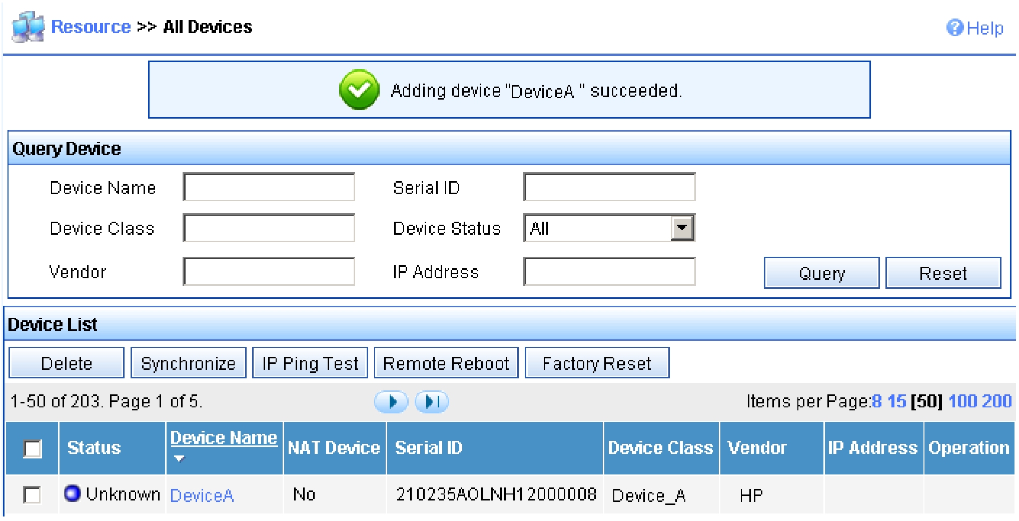
Figure 8 Adding a CPE



* CPE Name	DeviceA
* OUI	000FE2
* Serial ID	210231A95YH10C000045
CPE Class	HP Device_A
CPE Group	├DB_1

After the CPE is added successfully, a success message is displayed, as shown in Figure 9.

Figure 9 CPE added successfully



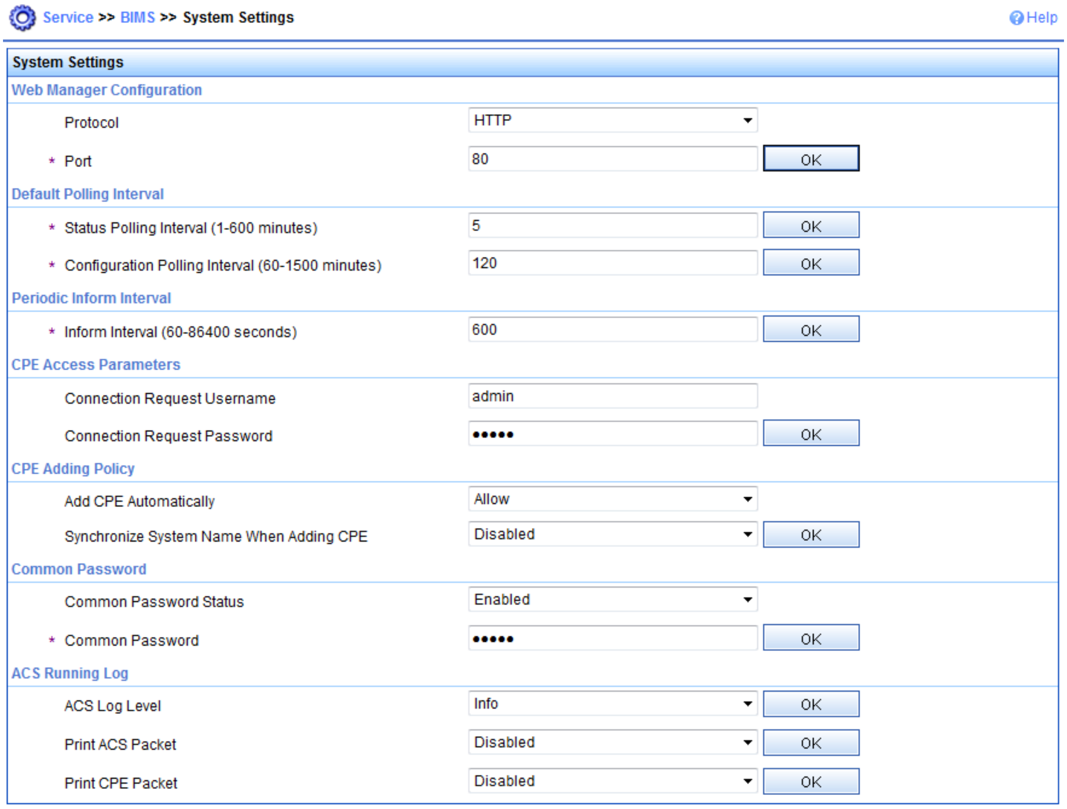
Adding device "DeviceA " succeeded.

1-50 of 203. Page 1 of 5.

□	Status	Device Name	NAT Device	Serial ID	Device Class	Vendor	IP Address	Operation
□	● Unknown	DeviceA	No	210235AOLNH12000008	Device_A	HP		

5. Configure the system settings of the ACS, as shown in Figure 10.

Figure 10 Configuring the system settings of the ACS



Service >> BIMS >> System Settings

Help

System Settings

Web Manager Configuration

Protocol	HTTP	
* Port	80	OK

Default Polling Interval

* Status Polling Interval (1-600 minutes)	5	OK
* Configuration Polling Interval (60-1500 minutes)	120	OK

Periodic Inform Interval

* Inform Interval (60-86400 seconds)	600	OK

CPE Access Parameters

Connection Request Username	admin	
Connection Request Password	•••••	OK

CPE Adding Policy

Add CPE Automatically	Allow	
Synchronize System Name When Adding CPE	Disabled	OK

Common Password

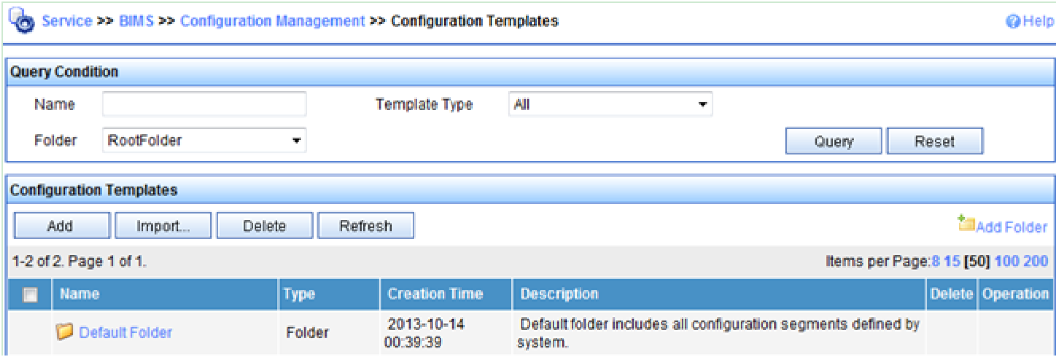
Common Password Status	Enabled	
* Common Password	•••••	OK

ACS Running Log

ACS Log Level	Info	OK
Print ACS Packet	Disabled	OK
Print CPE Packet	Disabled	OK

6. Add configuration templates and software library entries for the two classes of devices:
   a. Select **Service** > **BIMS** > **Configuration Management** > **Configuration Templates** from the navigation tree.

Figure 11 Configuring templates page



Service >> BIMS >> Configuration Management >> Configuration Templates

Help

Query Condition

Name		Template Type	All
Folder	RootFolder		

Query Reset

Configuration Templates

Add Import... Delete Refresh

Add Folder

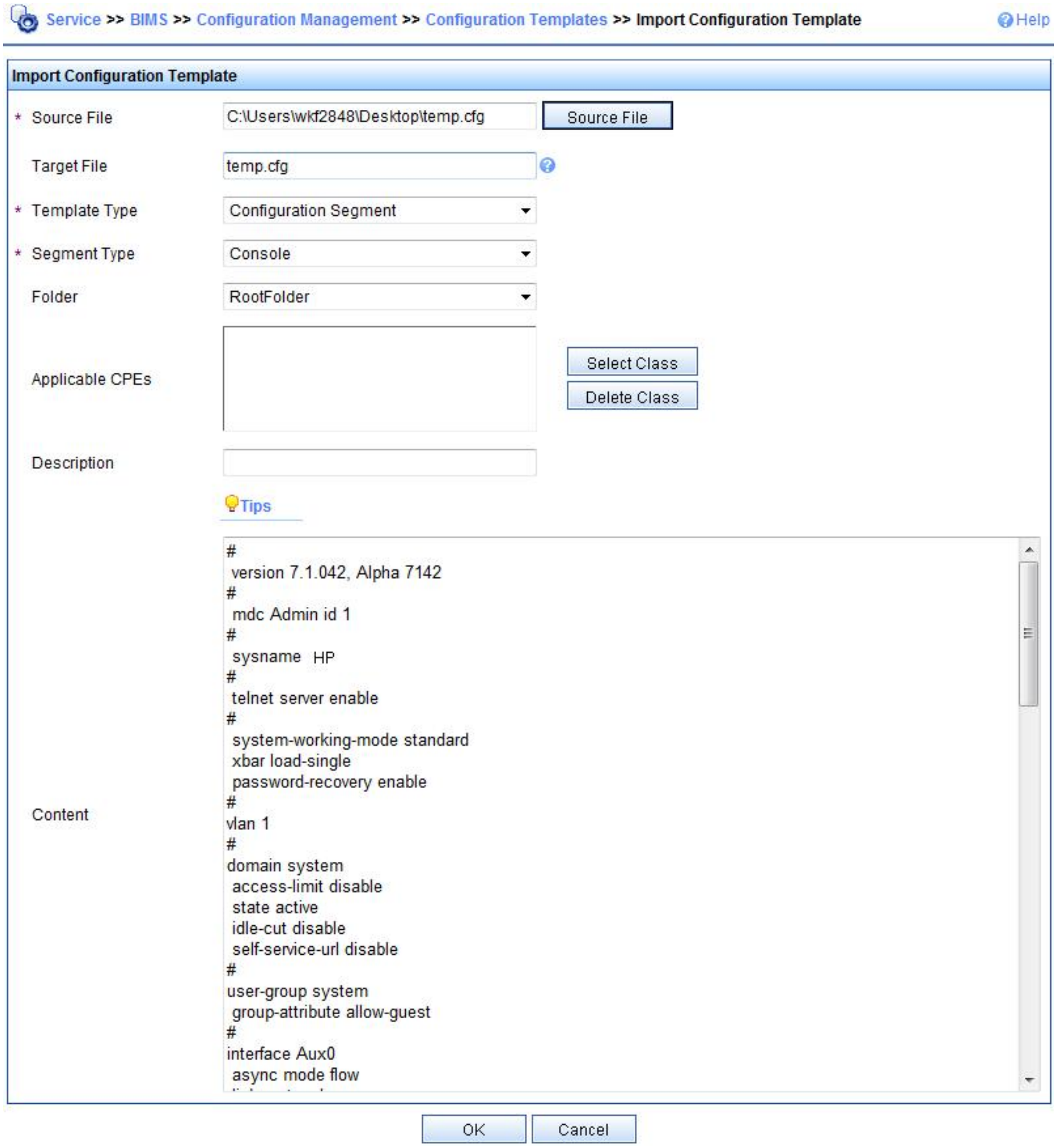
1-2 of 2. Page 1 of 1.

Items per Page: 8 15 [50] 100 200

□	Name	Type	Creation Time	Description	Delete	Operation
	Default Folder	Folder	2013-10-14 00:39:39	Default folder includes all configuration segments defined by system.		

   b. On the **Configuration Templates** page, click **Import...**.
   c. On the **Import Configuration Template** page, select configuration template settings for the **Device_A** device class, add the **Device_A** class to the **Applicable CPEs** pane, and then click **OK**.
   d. Repeat the previous two steps to configure a configuration template for equipment room B's device class.

Figure 12 Importing configuration template



Service >> BIMS >> Configuration Management >> Configuration Templates >> Import Configuration Template

Import Configuration Template

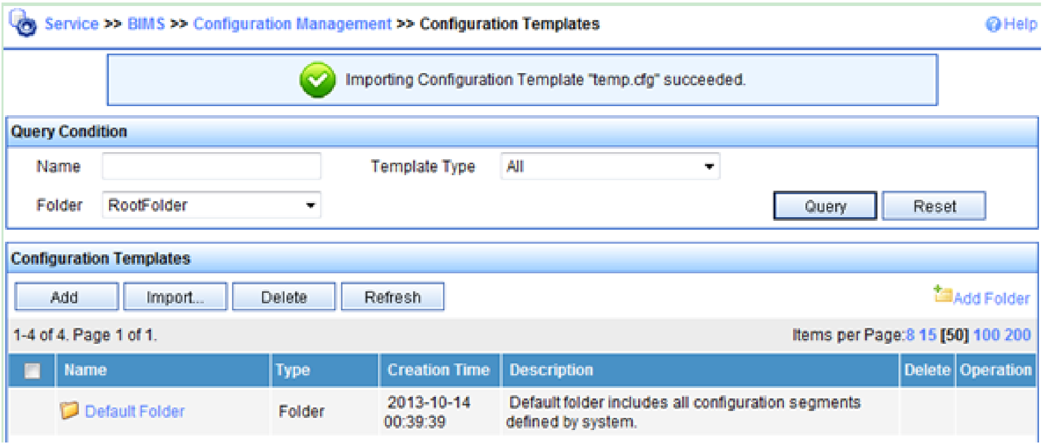
* Source File	C:\Users\wkf2848\Desktop\temp.cfg
Target File	temp.cfg
* Template Type	Configuration Segment
* Segment Type	Console
Folder	RootFolder
Applicable CPEs	
Description	

Content:

```
#
 version 7.1.042, Alpha 7142
#
 mdc Admin id 1
#
 sysname  HP
#
 telnet server enable
#
 system-working-mode standard
 xbar load-single
 password-recovery enable
#
vlan 1
#
domain system
 access-limit disable
 state active
 idle-cut disable
 self-service-url disable
#
user-group system
 group-attribute allow-guest
#
interface Aux0
 async mode flow
```

After the configuration template is added successfully, a success message is displayed, as shown in Figure 13.

Figure 13 Configuration templates



Service >> BIMS >> Configuration Management >> Configuration Templates

Importing Configuration Template "temp.cfg" succeeded.

Query Condition

Name		Template Type	All
Folder	RootFolder		

Configuration Templates

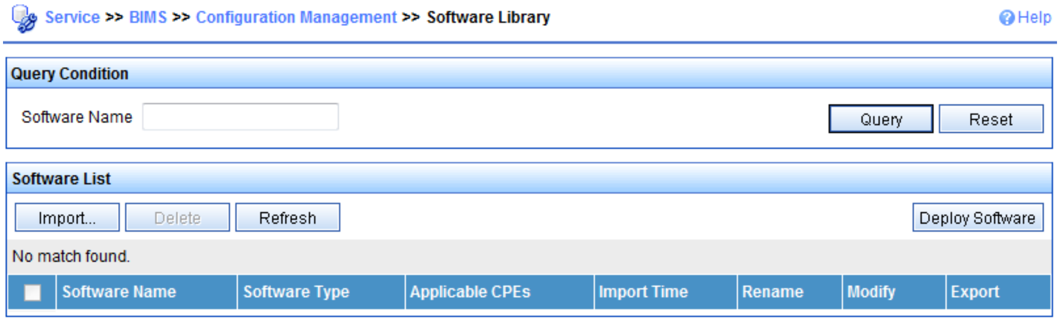
1-4 of 4. Page 1 of 1.

Items per Page:8 15 [50] 100 200

	Name	Type	Creation Time	Description	Delete	Operation
	Default Folder	Folder	2013-10-14 00:39:39	Default folder includes all configuration segments defined by system.		

e. Select **Service** > **BIMS** > **Configuration Management** > **Software Library** from the top navigation bar.

Figure 14 Configuring software library



Service >> BIMS >> Configuration Management >> Software Library

Help

Query Condition

Software Name

Query Reset

Software List

Import... Delete Refresh Deploy Software

No match found.

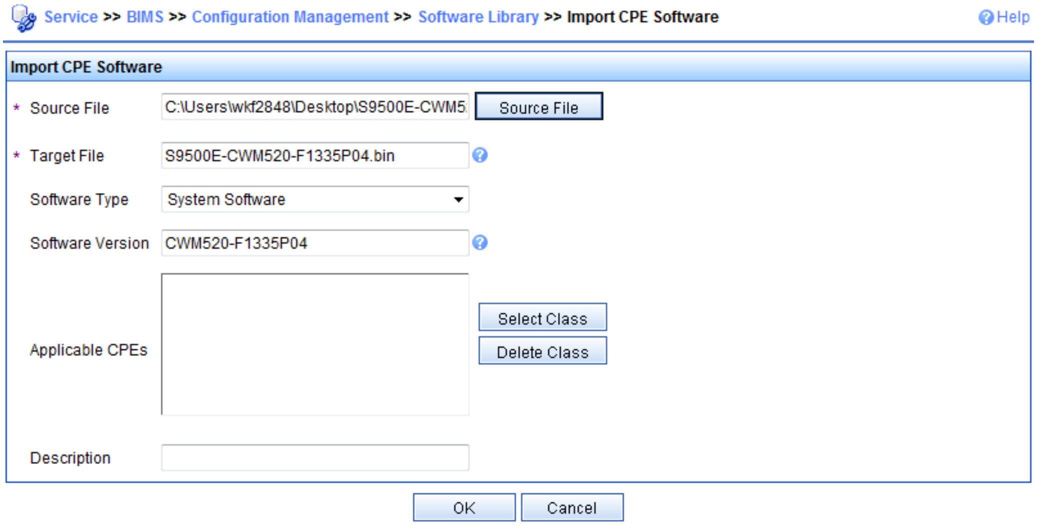
□	Software Name	Software Type	Applicable CPEs	Import Time	Rename	Modify	Export

f. On the **Software Library** page, click **Import…**.

g. On the **Import CPE Software** page, select the software images for the **Device_A** device class, add the **Device_A** class to the **Applicable CPEs** pane, and then click **OK**.

h. Repeat the previous two steps to configure a software library entry for equipment room B's device class.

Figure 15 Importing CPE software



Service >> BIMS >> Configuration Management >> Software Library >> Import CPE Software

Help

Import CPE Software

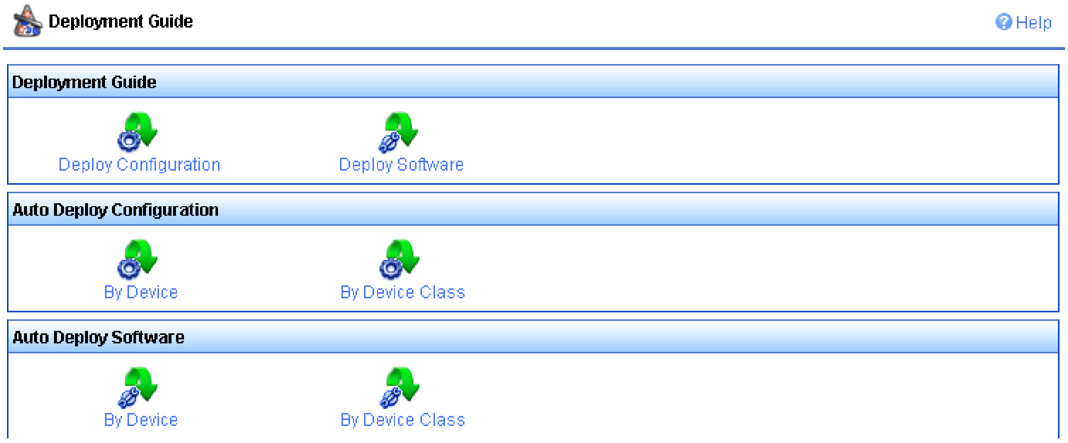
* Source File	C:\Users\wkf2848\Desktop\S9500E-CWM5
* Target File	S9500E-CWM520-F1335P04.bin
Software Type	System Software
Software Version	CWM520-F1335P04
Applicable CPEs	
Description	

7. Add auto-deployment tasks:

a. Select **Service** > **BIMS** > **Configuration Management** > **Deployment Guide** from the top navigation bar.

b. On the **Deployment Guide** page, click **By Device Class** in the **Auto Deploy Configuration** pane.

Figure 16 Deployment Guide



Deployment Guide

Help

Deployment Guide

Deploy Configuration

Deploy Software

Auto Deploy Configuration

By Device

By Device Class

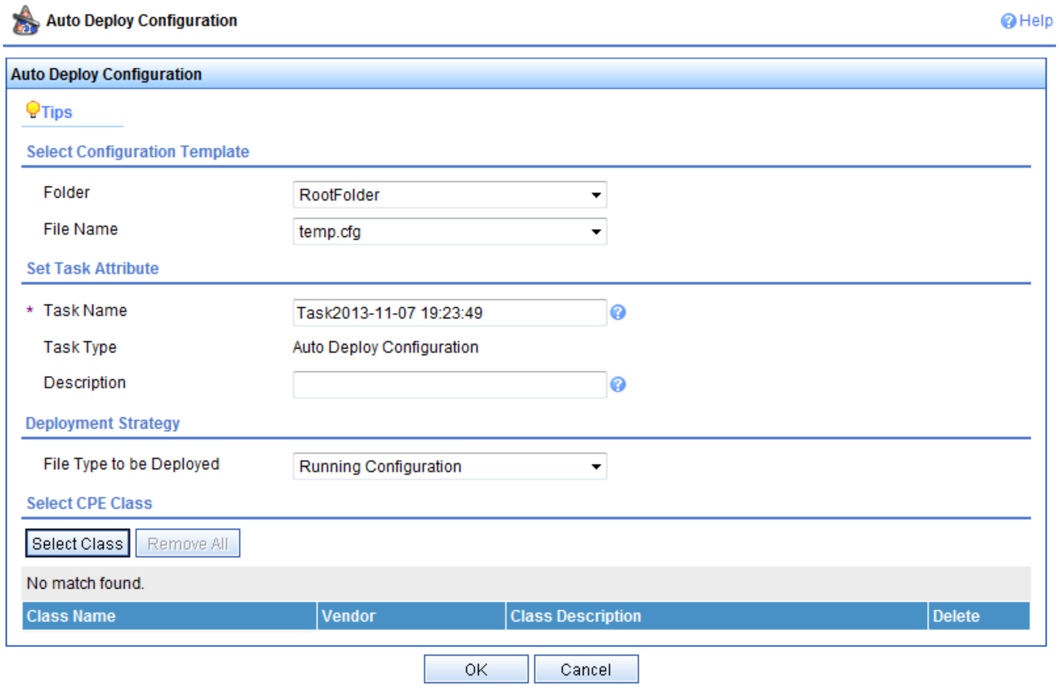
Auto Deploy Software

By Device

By Device Class

c. On the **Auto Deploy Configuration** page, click **Select Class**.

Figure 17 Configuring auto deployment



Auto Deploy Configuration

Help

Auto Deploy Configuration

Tips

Select Configuration Template

Folder: RootFolder

File Name: temp.cfg

Set Task Attribute

* Task Name: Task2013-11-07 19:23:49

Task Type: Auto Deploy Configuration

Description:

Deployment Strategy

File Type to be Deployed: Running Configuration

Select CPE Class

Select Class | Remove All

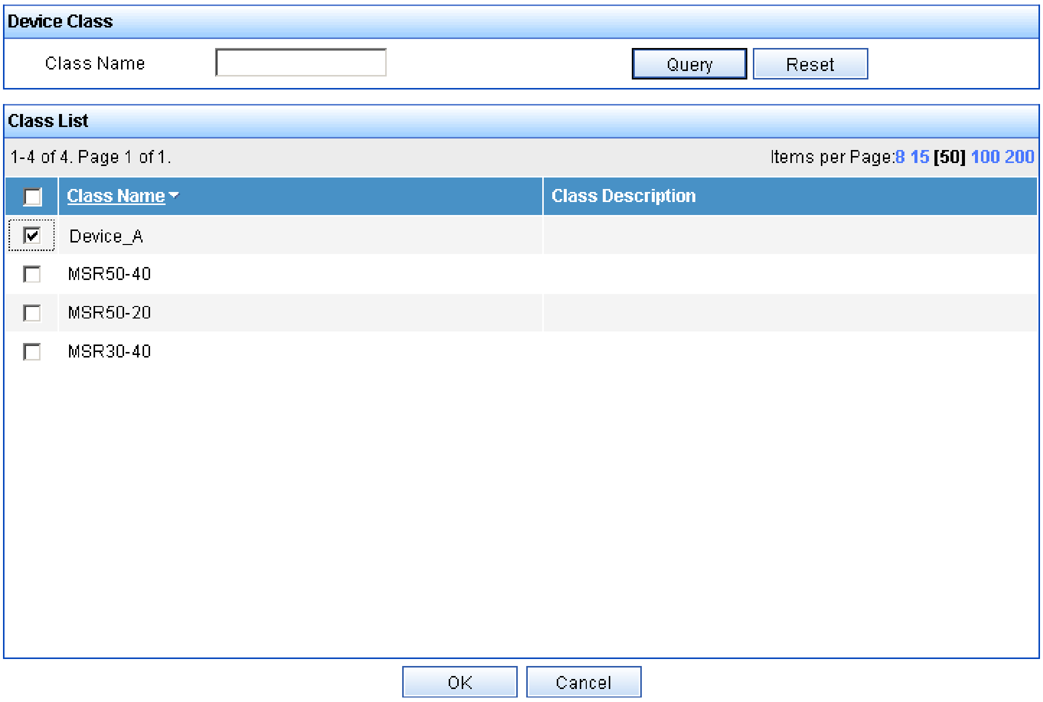
No match found.

Class Name	Vendor	Class Description	Delete

OK | Cancel

d. On the **Device Class** page, select **Device_A**, and then click **OK**.

Figure 18 Selecting device class



Device Class

Class Name [ ] Query Reset

Class List

1-4 of 4. Page 1 of 1. Items per Page:8 15 [50] 100 200

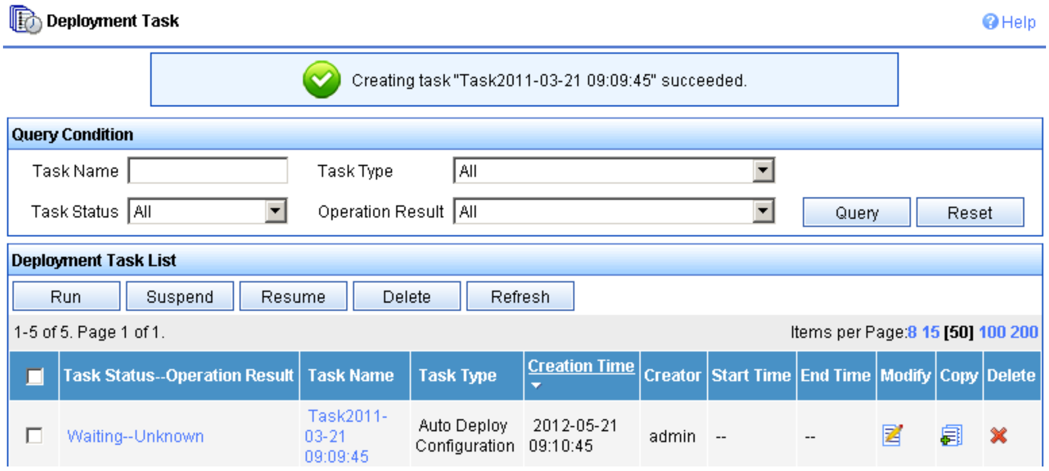
☐	Class Name	Class Description
☑	Device_A	
☐	MSR50-40	
☐	MSR50-20	
☐	MSR30-40	

OK Cancel

e. On the **Auto Deploy Configuration** page, click **OK**.

A success message is displayed, as shown in Figure 19.

Figure 19 Deployment task



Deployment Task Help

Creating task "Task2011-03-21 09:09:45" succeeded.

Query Condition

Task Name [ ] Task Type [All]

Task Status [All] Operation Result [All] Query Reset

Deployment Task List

Run Suspend Resume Delete Refresh

1-5 of 5. Page 1 of 1. Items per Page:8 15 [50] 100 200

☐	Task Status--Operation Result	Task Name	Task Type	Creation Time	Creator	Start Time	End Time	Modify	Copy	Delete
☐	Waiting--Unknown	Task2011-03-21 09:09:45	Auto Deploy Configuration	2012-05-21 09:10:45	admin	--	--	✎	⧉	✖

f. Add a deployment task for devices in equipment room B in the same way you add the deployment task for the devices in equipment room A.

Configuring the DHCP server

In this example, an HP device is operating as the DHCP server.

1. Configure an IP address pool to assign IP addresses and DNS server address to the CPEs. This example uses subnet 10.185.10.0/24 for IP address assignment.

   # Enable DHCP.

```
<DHCP_server> system-view
```

```
[DHCP_server] dhcp enable
```

# Enable DHCP server on VLAN-interface 1.

```
[DHCP_server] interface vlan-interface 1
[DHCP_server-Vlan-interface1] dhcp select server global-pool
[DHCP_server-Vlan-interface1] quit
```

# Exclude the DNS server address 10.185.10.60 and the ACS IP address 10.185.10.41 from dynamic allocation.

```
[DHCP_server] dhcp server forbidden-ip 10.185.10.41
[DHCP_server] dhcp server forbidden-ip 10.185.10.60
```

# Create DHCP address pool 0.

```
[DHCP_server] dhcp server ip-pool 0
```

# Assign subnet 10.185.10.0/24 to the address pool, and specify the DNS server address 10.185.10.60 in the address pool.

```
[DHCP_server-dhcp-pool-0] network 10.185.10.0 mask 255.255.255.0
[DHCP_server-dhcp-pool-0] dns-list 10.185.10.60
```

2. Configure DHCP Option 43 to contain the ACS URL, username, and password in hexadecimal format.

```
[DHCP_server-dhcp-pool-0] option 43 hex 0140 68747470 3A2F2F61 63732E64 61746162
6173653A 39303930 2F616373 20766963 6B792031 32333435
```

Configuring the DNS server

Map http://acs.database:9090/acs to http://10.185.1.41:9090/acs on the DNS server. For more information about DNS configuration, see DNS server documentation.

Connecting the CPEs to the network

# Connect the CPEs to the network, and then power on the CPEs. (Details not shown.)

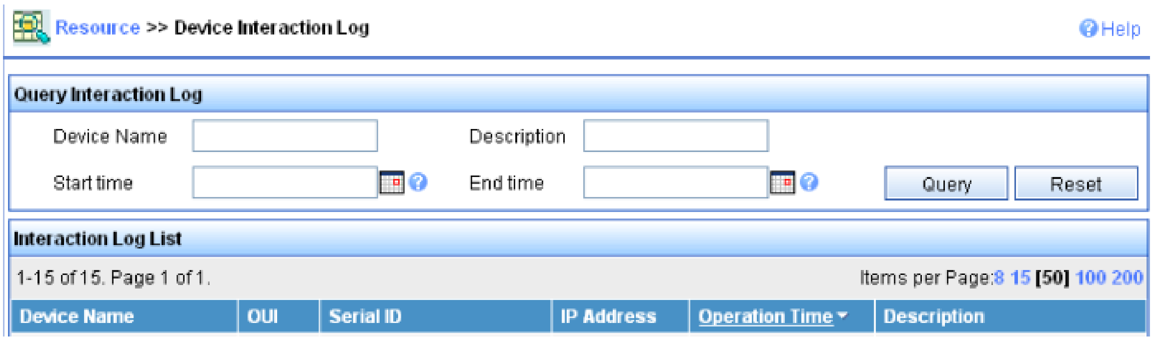
At startup, the CPEs obtain the IP address and ACS information from the DHCP server to initiate a connection to the ACS. After the connection is established, the CPEs interact with the ACS to complete autoconfiguration.

Verifying the configuration

Verify that the CPEs have obtained the correct configuration file from the ACS:

1. Select **Service** > **Resource** > **Device Interaction Log** from the top navigation bar.
2. On the **Device Interaction Log** page, verify that the configuration has been deployed on the CPEs.

Figure 20 Verifying the configuration deployment status



1-15 of 15. Page 1 of 1.

Items per Page:8 15 [50] 100 200

Device Name	OUI	Serial ID	IP Address	Operation Time	Description

Configuring NETCONF

Overview

Network Configuration Protocol (NETCONF) is an XML-based network management protocol with filtering capabilities. It provides programmable mechanisms to manage and configure network devices. Through NETCONF, you can configure device parameters, retrieve parameter values, and get statistics information.

In NETCONF messages, each data item is contained in a fixed element. This enables different devices of the same vendor to provide the same access method and the same result presentation method. For the devices of different vendors, XML mapping in NETCONF messages can achieve the same effect. For a network environment containing different devices regardless of vendors, you can develop a NETCONF-based NMS system to configure and manage devices in a simple and effective way.

NETCONF structure

NETCONF has four layers: content layer, operations layer, RPC layer, and transport protocol layer.

Table 27 NETCONF layers and XML layers

NETCONF layer	XML layer	Description
Content	Configuration data, status data, and statistics information	The content layer contains a set of managed objects, which can be configuration data, status data, and statistics information. For information about the operable data, see the NETCONF XML API reference for the switch.
Operations	<get>,<get-config>, <edit-config>…	The operations layer defines a set of base operations invoked as RPC methods with XML-encoded parameters. NETCONF base operations include data retrieval operations, configuration operations, lock operations, and session operations. For the device supported operations, see "Appendix A Supported NETCONF operations."
RPC	<rpc>,<rpc-reply>	The RPC layer provides a simple, transport-independent framing mechanism for encoding RPCs. The <rpc> and <rpc-reply> elements are used to enclose NETCONF requests and responses (data at the operations layer and the content layer).
Transport Protocol	• In non-FIPS mode: Console/Telnet/SSH/HTTP/HTTPS/TLS • In FIPS mode: Console/SSH/HTTPS/TLS	The transport protocol layer provides reliable, connection-oriented, serial data links. In non-FIPS mode, you can log in through Telnet, SSH, or the console port to perform NETCONF operations at the CLI. You can also log in through HTTP or HTTPS to perform NETCONF operations or perform NETCONF-over-SOAP operations. In FIPS mode, all login methods are the same as in non-FIPS mode except that you cannot use HTTP or Telnet.

NETCONF message format

NETCONF

IMPORTANT:

When configuring NETCONF in XML view, you must add the end mark "]]>]]>" at the end of an XML message. Otherwise, the device cannot identify the message. Examples in this chapter do not have this end mark. Do add it in actual operations.

All NETCONF messages are XML-based and comply with RFC 4741. Any incoming NETCONF messages must pass XML Schema check before it can be processed. If a NETCONF message fails XML Schema check, the device sends an error message to the client.

For information about the NETCONF operations supported by the device and the operable data, see the NETCONF XML API reference for the switch.

The following example shows a NETCONF message for getting all parameters of all interfaces on the device:

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id ="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
                 <Interface/>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

NETCONF over SOAP

All NETCONF-over-SOAP messages are XML-based and comply with RFC 4741. NETCONF messages are contained in the <Body> element of SOAP messages. NETCONF-over-SOAP messages also comply with the following rules:

- SOAP messages must use the SOAP Envelope namespaces.
- SOAP messages must use the SOAP Encoding namespaces.
- SOAP messages cannot contain the following information:
  - DTD reference.
  - XML processing instructions.

The following example shows a NETCONF-over-SOAP message for getting all parameters of all interfaces on the device:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <auth:Authentication env:mustUnderstand="1"
xmlns:auth="http://www.hp.com/netconf/base:1.0">
      <auth:AuthInfo>800207F0120020C</auth:AuthInfo>
    </auth:Authentication>
  </env:Header>
  <env:Body>
    <rpc message-id ="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <get-bulk>
        <filter type="subtree">
          <top xmlns="http://www.hp.com/netconf/data:1.0">
            <Ifmgr>
              <Interfaces>
                     <Interface/>
              </Interfaces>
            </Ifmgr>
          </top>
        </filter>
      </get-bulk>
    </rpc>
  </env:Body>
</env:Envelope>
```

How to use NETCONF

You can use NETCONF to manage and configure the device by using the methods in Table 28.

Table 28 NETCONF methods for configuring the device

Configuration tool	Login method	Remarks
CLI	• Console port • SSH • Telnet	To implement NETCONF operations, copy valid NETCONF messages to the CLI in XML view. This method is suitable for R&D and test purposes.
Custom configuration tool	N/A	To use this method, you must enable NETCONF over SOAP. By default, the device cannot interpret Custom configuration tools' URLs. For the device to interpret these URLs, you must encode the NETCONF messages sent from a custom configuration tool in SOAP.

Protocols and standards

- RFC 3339, *Date and Time on the Internet: Timestamps*
- RFC 4741, *NETCONF Configuration Protocol*
- RFC 4742, *Using the NETCONF Configuration Protocol over Secure SHell (SSH)*
- RFC 4743, *Using NETCONF over the Simple Object Access Protocol (SOAP)*
- RFC 5277, *NETCONF Event Notifications*
- RFC 5381, *Experience of Implementing NETCONF over SOAP*
- RFC 5539, *NETCONF over Transport Layer Security (TLS)*

- RFC 6241, *Network Configuration Protocol*

FIPS compliance

The device supports the FIPS mode that complies with NIST FIPS 140-2 requirements. Support for features, commands, and parameters might differ in FIPS mode (see *Security Configuration Guide*) and non-FIPS mode.

NETCONF configuration task list

Task at a glance
(Optional.) Enabling NETCONF over SOAP
(Optional.) Enabling NETCONF over SSH
(Required.) Establishing a NETCONF session
(Optional.) Subscribing to event notifications
(Optional.) Locking/unlocking the configuration
(Optional.) Performing the get/get-bulk operation
(Optional.) Performing the get-config/get-bulk-config operation
(Optional.) Performing the edit-config operation
(Optional.) Saving, rolling back, and loading the configuration
(Optional.) Filtering data
(Optional.) Performing CLI operations through NETCONF
(Optional.) Retrieving NETCONF session information
(Optional.) Terminating another NETCONF session
(Optional.) Returning to the CLI

Enabling NETCONF over SOAP

NETCONF messages can be encapsulated into SOAP messages and transmitted over HTTP and HTTPS. After enabling NETCONF over SOAP, you can develop a configuration interface to perform NETCONF operations.

To enable NETCONF over SOAP:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. Enable NETCONF over SOAP.	- Enable NETCONF over SOAP over HTTP (not available in FIPS mode): **netconf soap http enable** - Enable NETCONF over SOAP over HTTPS: **netconf soap https enable**	By default, NETCONF over SOAP is disabled.

Enabling NETCONF over SSH

This feature allows users to use a client to perform NETCONF operations on the device through a NETCONF-over-SSH connection.

To enable NETCONF over SSH:

Step	Command	Remark
1. Enter system view.	**system-view**	N/A
2. Enable NETCONF over SSH.	**netconf ssh server enable**	By default, NETCONF over SSH is disabled.
3. Specify a port to listen for NETCONF-over-SSH connections.	**netconf ssh server port** *port number*	By default, port 830 listens for NETCONF-over-SSH connections.

Establishing a NETCONF session

A client must send a hello message to a device to finish capabilities exchange before the device processes other requests from the client.

The device supports a maximum of 32 NETCONF sessions. If the upper limit is reached, new NETCONF users cannot access the device.

Entering xml view

Task	Command	Remarks
Enter XML view.	**xml**	Available in user view.

Exchanging capabilities

After you enter XML view, the client and the device exchange their capabilities before you can perform subsequent operations. The device automatically advertises its NETCONF capabilities to the client in a hello message as follows:

```
<?xml version="1.0" encoding="UTF-8"?><hello
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><capabilities><capability>urn:ietf:pa
rams:netconf:base:1.1</capability><capability>urn:ietf:params:netconf:writable-runnin
```

```
g</capability><capability>urn:ietf:params:netconf:capability:notification:1.0</capabi
lity><capability>urn:ietf:params:netconf:capability:validate:1.1</capability><capabil
ity>urn:ietf:params:netconf:capability:interleave:1.0</capability><capability>urn:hp:
params:netconf:capability:hp-netconf-ext:1.0</capability></capabilities><session-id>1
</session-id></hello>]]>]]>
```

Where:

- The <capabilities> parameter represents the capabilities supported by the device.
- The <session-id> parameter represents the unique ID assigned to the current session.

After receiving the hello message from the device, copy the following message to notify the device of the capabilities (user-configurable) supported by the client:

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
     capability-set
   </capability>
 </capabilities>
</hello>
```

Where, *capability-set* represents the capabilities supported by the client. Use a pair of <capability> and </capability> tags to enclose a capability.

Subscribing to event notifications

After you subscribe to event notifications, the device sends event notifications to the NETCONF client when a subscribed event takes place on the device. The notifications include the code, group, severity, start time, and description of the events.

A subscription takes effect only on the current session. If the session is terminated, the subscription is automatically canceled.

You can send multiple subscription messages to subscribe to notification of multiple events.

Subscription procedure

# Copy the following message to the client to complete the subscription:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <create-subscription  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
         <stream>NETCONF</stream>
         <filter>
             <event xmlns="http://www.hp.com/netconf/event:1.0">
                 <Code>code</Code>
                 <Group>group</Group>
                 <Severity>severity</Severity>
             </event>
         </filter>
         <startTime>start-time</startTime>
         <stopTime>stop-time</stopTime>
     </create-subscription>
```

```
</rpc>
```

Where:

- The <stream> parameter represents the event stream type supported by the device. Only NETCONF is supported.
- The <event> parameter represents an event to which you have subscribed.
- The <code> parameter represents a mnemonic symbol.
- The <group> parameter represents the module name.
- The <severity> parameter represents the severity level of the event.
- The <start-time> parameter represents the start time of the subscription.
- The <stop-time> argument represents the end time of the subscription.

After receiving the subscription request from the client, the device returns a response in the following format if the subscription is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply >
```

If the subscription fails, the device returns an error message in the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<rpc-error>
   <error-type>error-type</error-type>
   <error-tag>error-tag</error-tag>
   <error-severity>error-severity</error-severity>
   <error-message xml:lang="en">error-message</error-message>
</rpc-error>
</rpc-reply>
```

For more information about error messages, see RFC 4741.

Example for subscribing to event notifications

Network requirements

Configure a client to subscribe to all events with no time limitation. After the subscription is successful, all events on the device are sent to the client before the session between the device and client is terminated.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Subscribe to all events with no time limitation.

```
<rpc message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <create-subscription xmlns ="urn:ietf:params:xml:ns:netconf:notification:1.0">
        <stream>NETCONF</stream>
    </create-subscription>
</rpc>
```

Verifying the configuration

# If the client receives the following response, the subscription is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <ok/>
</rpc-reply>
```

# If fan 1 on the device encounters problems, the device sends the following text to the client that has subscribed to all events:

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2011-01-04T12:30:46</eventTime>
    <event xmlns="http://www.hp.com/netconf/event:1.0">
        <Group>DEV</Group>
        <Code>FAN_DIRECTION_NOT_PREFERRED</Code>
        <Slot>6</Slot>
        <Severity>Alert</Severity>
        <context>Fan 1 airflow direction is not preferred on slot 6, please check
it.</context>
    </event>
</notification>
```

# When another client (192.168.100.130) logs in to the device, the device sends a notification to the client that has subscribed to all events:

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2011-01-04T12:30:52</eventTime>
    <event xmlns="http://www.hp.com/netconf/event:1.0">
        <Group>SHELL</Group>
        <Code>SHELL_LOGIN</Code>
        <Slot>6</Slot>
        <Severity>Notification</Severity>
        <context>VTY logged in from 192.168.100.130.</context>
    </event>
</notification>
```

Locking/unlocking the configuration

The device supports a maximum of 32 NETCONF sessions. A maximum of 32 users can simultaneously manage and monitor the device using NETCONF. During device configuration and maintenance or network troubleshooting, a user can lock the configuration to prevent other users from changing it. After that, only the user holding the lock can change the configuration, and other users can only read the configuration.

In addition, only the user holding the lock can release the lock. After the lock is released, other users can change the current configuration or lock the configuration. If the session of the user that holds the lock is terminated, the system automatically releases the lock.

Locking the configuration

# Copy the following text to the client to lock the configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <lock>
      <target>
        <running/>
      </target>
    </lock>
  </rpc>
```

After receiving the lock request, the device returns a response in the following format if the lock operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <ok/>
</rpc-reply>
```

Unlocking the configuration

# Copy the following text to the client to unlock the configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <unlock>
      <target>
        <running/>
      </target>
    </unlock>
  </rpc>
```

After receiving the unlock request, the device returns a response in the following format if the unlock operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

Example for locking the configuration

Network requirements

Lock the device configuration so that other users cannot change the device configuration.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Lock the configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <lock>
      <target>
        <running/>
      </target>
    </lock>
  </rpc>
```

Verifying the configuration

If the client receives the following response, the lock operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

If another client sends a lock request, the device returns the following response:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<rpc-error>
  <error-type>protocol</error-type>
  <error-tag>lock-denied</error-tag>
  <error-severity>error</error-severity>
  <error-message xml:lang="en">Lock failed because the NETCONF lock is held by another
session.</error-message>
  <error-info>
    <session-id>1</session-id>
  </error-info>
  </rpc-error>
</rpc-reply>
```

The output shows that the lock operation failed because the client with session ID 1 held the lock, and only the client holding the lock can release the lock.

Performing service operations

You can use NETCONF to perform service operations on the device, such as retrieving and modifying the specified information. The basic operations include get, get-bulk, get-config, get-bulk-config, and edit-config, which are used to retrieve all data, retrieve configuration data, and edit the data of the specified module. For more information, see the NETCONF XML API reference for the switch.

Performing the get/get-bulk operation

The get operation is used to retrieve device configuration and state information that match the conditions. In some cases, this operation leads to inefficiency.

The get-bulk operation is used to retrieve a number of data entries starting from the data entry next to the one with the specified index. One data entry contains a device configuration entry and a state information entry. The data entry quantity is defined by the *count* attribute, and the index is specified by the *index* attribute. The returned output does not include the index information. If you do not specify the *index* attribute, the index value starts with 1 by default.

If either of the following conditions occurs, the get-bulk operation retrieves all the rest data entries starting from the data entry next to the one with the specified index:

- You do not specify the *count* attribute.
- The number of matched data entries is less than the value of the *count* attribute.

# Copy the following text to the client to perform the get operation:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <getoperation>
    <filter>
      <top xmlns=" http://www.hp.com/netconf/data:1.0">
          Specify the module, submodule, table name, and column name
      </top>
    </filter>
  </getoperation>
</rpc>
```

Where, the <getoperation> parameter can be <get> or <get-bulk>. The <filter> element is used to filter data, and it can contain module name, submodule name, table name, and column name.

- If the module name and the submodule name are not provided, the operation retrieves the data for all modules and submodules. If a module name or a submodule name is provided, the operation retrieves the data for the specified module or submodule.
- If the table name is not provided, the operation retrieves the data for all tables. If a table name is provided, the operation retrieves the data for the specified table.
- If only the index column is provided, the operation retrieves the data for all columns. If the index column and other columns are provided, the operation retrieves the data for the index column and the specified columns.

The <get> and <get-bulk> messages are similar. A <get-bulk> message carries the *count* and *index* attributes. The following is a <get-bulk> message example:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="http://www.hp.com/netconf/base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0"
xmlns:base="http://www.hp.com/netconf/base:1.0">
        <Syslog>
            <Logs xc:count="5">
                   <Log>
                      <Index>10</Index>
                    </Log>
             </Logs>
        </Syslog>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

Where, the *count* attribute complies with the following rules:

- The *count* attribute can be placed in the module node and table node. In other nodes, it cannot be resolved.
- When the *count* attribute is placed in the module node, a descendant node inherits this *count* attribute if the descendant node does not contain the *count* attribute.

Verifying the configuration

After receiving the get-bulk request, the device returns a response in the following format if the operation is successful:

```
<?xml version="1.0"?>
<rpc-reply message-id="100"
           xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
     Device state and configuration data
  </data>
</rpc-reply>
```

Performing the get-config/get-bulk-config operation

The get-config and get-bulk-config operations are used to retrieve all non-default configurations, which are configured by means of CLI and MIB. The <get-config> and <get-bulk-config> messages can contain the <filter> element for filtering data.

The <get-config> and <get-bulk-config> messages are similar. The following is a <get-config> message example:

```
<?xml version="1.0"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
```

```
      <running/>
    </source>
    <filter>
      <top xmlns="http://www.hp.com/netconf/config:1.0">
          Specify the module name, submodule name, table name, and column name
      </top>
    </filter>
  </get-config>
</rpc>
```

Verifying the configuration

After receiving the get-config request, the device returns a response in the following format if the operation is successful:

```
<?xml version="1.0"?>
<rpc-reply message-id="100"   xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    All data matching the specified filter
  </data>
</rpc-reply>
```

Performing the edit-config operation

The edit-config operation supports the following operation attributes: merge, create, replace, remove, delete, default-operation, error-option, and test-option. For more information about these attributes, see "Appendix A Supported NETCONF operations."

# Copy the following text to perform the <edit-config> operation:

```
<?xml version="1.0"?>
<rpc message-id="100"  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running></running></target>
<error-option>
   Default operation when an error occurs
</error-option>
    <config>
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        Specify the module name, submodule name, table name, and column name
      </top>
    </config>
  </edit-config>
</rpc>
```

After receiving the edit-config request, the device returns a response in the following format if the operation is successful:

```
<?xml version="1.0">
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

# Perform the get operation to verify that the current value of the parameter is the same as the value specified through the edit-config operation. (Details not shown.)

All-module configuration data retrieval example

Network requirements

Retrieve configuration data for all modules.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Retrieve configuration data for all modules.

```
<rpc message-id="100"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
  </get-config>
</rpc>
```

Verifying the configuration

If the client receives the following text, the get-config operation is successful:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <data>
        <top xmlns="http://www.hp.com/netconf/config:1.0">
            <Ifmgr>
                <Interfaces>
                    <Interface>
                        <IfIndex>1307</IfIndex>
                        <Shutdown>1</Shutdown>
                    </Interface>
                    <Interface>
                        <IfIndex>1308</IfIndex>
                        <Shutdown>1</Shutdown>
                    </Interface>
                    <Interface>
                        <IfIndex>1309</IfIndex>
                        <Shutdown>1</Shutdown>
                    </Interface>
                    <Interface>
                        <IfIndex>1311</IfIndex>

                            <VlanType>2</VlanType>

                    </Interface>
                    <Interface>
                        <IfIndex>1313</IfIndex>

                            <VlanType>2</VlanType>

                    </Interface>
                </Interfaces>
            </Ifmgr>
            <Syslog>
                <LogBuffer>
                    <BufferSize>120</BufferSize>
                </LogBuffer>
            </Syslog>
            <System>
                <Device>
                    <SysName>HP</SysName>
                    <TimeZone>
                        <Zone>+11:44</Zone>
                        <ZoneName>beijing</ZoneName>
                    </TimeZone>
                </Device>
            </System>
        </top>
    </data>
</rpc-reply>
```

Syslog configuration data retrieval example

Network requirements

Retrieve configuration data for the Syslog module.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Retrieve configuration data for the Syslog module.

```
<rpc message-id="100"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        <Syslog/>
      </top>
    </filter>
  </get-config>
</rpc>
```

Verifying the configuration

If the client receives the following text, the get-config operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <data>
        <top xmlns="http://www.hp.com/netconf/config:1.0">
            <Syslog>
                    <LogBuffer>
                        <BufferSize>120</BufferSize>
                    </LogBuffer>
            </Syslog>
        </top>
    </data>
</rpc-reply>
```

Example for retrieving a data entry for the interface table

Network requirements

Retrieve a data entry for the interface table.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>urn:ietf:params:netconf:base:1.0</capability>
    </capabilities>
</hello>
```

Retrieve a data entry for the interface table.

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get-bulk>
        <filter type="subtree">
            <top xmlns="http://www.hp.com/netconf/data:1.0"
xmlns:web="http://www.hp.com/netconf/base:1.0">
                <Ifmgr>
                    <Interfaces web:count="1">
                    </Interfaces>
                </Ifmgr>
            </top>
        </filter>
    </get-bulk>
</rpc>
```

Verifying the configuration

If the client receives the following text, the get-bulk operation is successful:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
    <top xmlns="http://www.hp.com/netconf/data:1.0">
      <Ifmgr>
        <Interfaces>
          <Interface>
            <IfIndex>3</IfIndex>
            <Name>Ten-GigabitEthernet1/0/2</Name>
            <AbbreviatedName>XGE1/0/2</AbbreviatedName>
            <PortIndex>3</PortIndex>
            <ifTypeExt>22</ifTypeExt>
            <ifType>6</ifType>
            <Description>Ten-GigabitEthernet 1/0/2 Interface</Description>
            <AdminStatus>2</AdminStatus>
            <OperStatus>2</OperStatus>
            <ConfigSpeed>0</ConfigSpeed>
            <ActualSpeed>100000</ActualSpeed>
            <ConfigDuplex>3</ConfigDuplex>
            <ActualDuplex>1</ActualDuplex>
          </Interface>
        </Interfaces>
      </Ifmgr>
    </top>
  </data>
</rpc-reply>
```

Example for changing the value of a parameter

Network requirements

Change the log buffer size for the Syslog module to 512.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>urn:ietf:params:netconf:base:1.0</capability>
    </capabilities>
</hello>
```

# Change the log buffer size for the Syslog module to 512.

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0">
<edit-config>
    <target>
        <running/>
    </target>
    <config>
        <top xmlns="http://www.hp.com/netconf/config:1.0" web:operation="merge">
            <Syslog>
                <LogBuffer>
                    <BufferSize>512</BufferSize>
                </LogBuffer>
            </Syslog>
        </top>
    </config>
</edit-config>
</rpc>
```

Verifying the configuration

If the client receives the following text, the edit-config operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
```

Saving, rolling back, and loading the configuration

Use NETCONF to save, roll back, or load the configuration.

Saving the configuration

# Copy the following text to the client to save the device configuration to the specified file:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<save>
 <file>Specify the configuration file name</file>
</save>
</rpc>
```

The name of the specified configuration file must start with the storage media name and end with the extension (**.cfg**). The total length of the save path and file name must be no more than 191 characters. If the text includes the file column, you must specify the file name and the specified file will be used as the next-startup configuration file. If the text does not include the file column, the configuration is automatically saved to the default main next-startup configuration file.

After receiving the save request, the device returns a response in the following format if the save operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

Rolling back the configuration

# Copy the following text to the client to roll back the configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<rollback>
 <file>Specify the configuration file name</file>
</rollback>
</rpc>
```

The name of the specified configuration file must start with the storage media name and end with the extension (**.cfg**). The total length of the save path and file name must be no more than 191 characters.

After receiving the rollback request, the device returns a response in the following format if the rollback operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

Loading the configuration

After you perform the load operation, the loaded configurations are merged into the current configuration:

- New configurations are directly loaded.

- Configurations that already exist in the current configuration are replaced by those loaded from the configuration file.

# Copy the following text to the client to load a configuration file for the device:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <load>
     <file>Specify the configuration file name</file>
  </load>
</rpc>
```

The name of the specified configuration file must start with the storage media name and end with the extension (**.cfg**). The total length of the save path and file name must be no more than 191 characters.

After receiving the load request, the device returns a response in the following format if the load operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

Example for saving the configuration

Network requirements

Save the current configuration to the configuration file **my_config.cfg**.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Save the configuration of the device to the configuration file **my_config.cfg**.

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <save>
   <file>my_config.cfg</file>
</save>
</rpc>
```

Verifying the configuration

If the client receives the following response, the save operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

Filtering data

You can define a filter with the <filter> element to filter information when you perform a get, get-bulk, get-config, or get-bulk-config operation. Data filtering mechanisms include full match, regular expression match, and conditional match.

- Full match

  You can specify an element value in an XML message to implement full match. If multiple element values are provided, the system returns the data that matches all the specified values.

  # Copy the following text to the client to retrieve the configuration data of all interfaces in UP state:

```
<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <AdminStatus>2</AdminStatus>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>
```

  You can also specify an attribute name that is the same as a column name of the current table at the row to implement full match. The system returns only configuration data that matches this attribute name. The XML message equivalent to the above element-value-based full match is as follows:

```
<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0"xmlns:data="
http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface data:AdminStatus="2"/>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>
```

The above examples show that both element-value-based full match and attribute-name-based full match can retrieve the same configuration data for all interfaces in up state.

- Regular expression match

  To implement a complex data filtering with characters, you can add a *regExp* attribute for a specific element.

  # Copy the following text to the client to retrieve the descriptions of interfaces, of which all the characters must be upper-case letters from A to Z:

```
<rpc message-id="1-0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:hp="http://www.hp.com/netconf/base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <Description hp:regExp="[A-Z]*"/>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-config>
</rpc>
```

- Conditional match

  To implement a complex data filtering with digits and character strings, you can add a *match* attribute for a specific element. Table 29 lists the conditional match operators.

Table 29 Conditional match operators

Operation	Operator	Remarks
More than	match="more:*value*"	More than the specified *value*. The supported data types include date, digit, and character string.
Less than	match="less:*value*"	Less than the specified value. The supported data types include date, digit, and character string.
Not less than	match="notLess:*value*"	Not less than the specified *value*. The supported data types include date, digit, and character string.
Not more than	match="notMore:*value*"	Not more than the specified *value*. The supported data types include date, digit, and character string.
Equal	match="equal:*value*"	Equal to the specified *value*. The supported data types include date, digit, character string, OID, and BOOL.
Not equal	match="notEqual:*value*"	Not equal to the specified *value*. The supported data types include date, digit, character string, OID, and BOOL.

Operation	Operator	Remarks
Include	match="include:*string*"	Includes the specified *string*. The supported data types include only character string.
Not include	match="exclude:*string*"	Excludes the specified *string*. The supported data types include only character string.
Start with	match="startWith:*string*"	Starts with the specified *string*. The supported data types include character string and OID.
End with	match="endWith:*string*"	Ends with the specified *string*. The supported data types include only character string.

# Copy the following text to the client to retrieve extension information about the entity of which the CPU usage is more than 50%:

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:hp="http://www.hp.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Device>
          <ExtPhysicalEntities>
            <Entity>
                <CpuUsage hp:match="more:50"></CpuUsage>
            </Entity>
          </ExtPhysicalEntities>
        </Device>
      </top>
    </filter>
  </get>
</rpc>
```

Example for filtering data with regular expression match

Network requirements

Retrieve all data including colons in the **Description** column of the Interfaces table under the Ifmgr module.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

Retrieve all data including colons in the **Description** column of the Interfaces table under the Ifmgr module.

```
<?xml version="1.0"?>
<rpc message-id="100"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:reg="http://www.hp.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
                <Description  reg:regExp=":"/>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>
```

Verifying the configuration

If the client receives the following text, the operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:reg="http://www.hp.com/netconf/base:1.0" message-id="100">
    <data>
        <top xmlns="http://www.hp.com/netconf/data:1.0">
            <Ifmgr>
                <Interfaces>
                    <Interface>
                        <IfIndex>2681</IfIndex>
                        <Description>Ten-GigabitEthernet1/0/1 Interface</Description>
                    </Interface>
                    <Interface>
                        <IfIndex>2682</IfIndex>
                        <Description>Ten-GigabitEthernet1/0/2 Interface</Description>
                    </Interface>
                    <Interface>
                        <IfIndex>2683</IfIndex>
                        <Description>Ten-GigabitEthernet1/0/3 Interface</Description>
                    </Interface>
                    <Interface>
                        <IfIndex>2684</IfIndex>
                        <Description>Ten-GigabitEthernet1/0/4 Interface</Description>
                    </Interface>
                 <Interface>
            </Ifmgr>
        </top>
    </data>
</rpc-reply>
```

Example for filtering data by conditional match

Network requirements

Retrieve data in the **Name** column with the ifindex value not less than 5000 in the Interfaces table under the Ifmgr module.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Retrieve data in the **Name** column with the ifindex value not less than 5000 in the Interfaces table under the Ifmgr module.

```
<rpc message-id="100"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="http://www.hp.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
                <IfIndex nc:match="more:5000"/>
                <Name/>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>
```

Verifying the configuration

If the client receives the following text, the operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="http://www.hp.com/netconf/base:1.0" message-id="100">
    <data>
        <top xmlns="http://www.hp.com/netconf/data:1.0">
```

```
                <Ifmgr>
                    <Interfaces>
                        <Interface>
                            <IfIndex>7241</IfIndex>
                            <Name>NULL0</Name>
                        </Interface>
                    </Interfaces>
                </Ifmgr>
            </top>
        </data>
</rpc-reply>
```

Performing CLI operations through NETCONF

You can enclose command lines in XML messages to configure the device.

Configuration procedure

# Copy the following text to the client to execute the commands:

```
<?xml version="1.0" encoding="UTF-8"?>
   <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <CLI>
        <Execution>
          Commands
        </Execution>
      </CLI>
</rpc>
```

The <Execution> element can contain multiple commands, with one command on one line.

After receiving the CLI operation request, the device returns a response in the following format if the CLI operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution>
      <![CDATA[Responses to the commands]]>
    </Execution>
  </CLI>
</rpc-reply>
```

CLI operation example

Configuration requirements

Send the **display current-configuration** command to the device.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Copy the following text to the client to execute the **display current-configuration** command:

```
<?xml version="1.0" encoding="UTF-8"?>
   <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <CLI>
        <Execution>
          display current-configuration
        </Execution>
      </CLI>
</rpc>
```

Verifying the configuration

If the client receives the following text, the operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution><![CDATA[
<Sysname>display current-configuration
#
 version 7.1.045, ESS 2305
#
 sysname ASBR2
#
 telnet server enable
#
 irf mac-address persistent timer
 irf auto-update enable
 undo irf link-delay
 irf member 1 priority 1
   ]]>
   </Execution>
  </CLI>
</rpc-reply>
```

Retrieving NETCONF session information

You can use the get-sessions operation to retrieve NETCONF session information of the device.

# Copy the following message to the client to retrieve NETCONF session information from the device:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get-sessions/>
</rpc>
```

After receiving the get-sessions request, the device returns a response in the following format if the get-sessions operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions>
    <Session>
      <SessionID>Configuration session ID </SessionID>
      <Line>line information</Line>
      <UserName>Name of the user creating the session</UserName>
      <Since>Time when the session was created</Since>
      <LockHeld>Whether the session holds a lock</LockHeld>
    </Session>
  </get-sessions>
</rpc-reply>
```

For example, to get NETCONF session information:

# Enter XML view.

```
<Sysname> xml
```

# Copy the following message to the client to exchange capabilities with the device:

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Copy the following message to the client to get the current NETCONF session information on the device:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get-sessions/>
</rpc>
```

If the client receives a message as follows, the operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <get-sessions>
        <Session>
            <SessionID>1</SessionID>
            <Line>vty0</Line>
            <UserName></UserName>
            <Since>2013-01-05T00:24:57</Since>
            <LockHeld>false</LockHeld>
        </Session>
    </get-sessions>
```

```
</rpc-reply>
```

The output shows an existing NETCONF session with session ID as 1. The login user type is vty0, the login time is 2013-01-05T00:24:57, and the user does not hold the lock of the configuration.

Terminating another NETCONF session

NETCONF allows one client to terminate the NETCONF session of another client. The client whose session is terminated returns to user view.

# Copy the following message to the client to terminate the specified NETCONF session:

```
<rpc message-id="101"    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <kill-session>
      <session-id>
        Specified session-ID
      </session-id>
     </kill-session>
   </rpc>
```

After receiving the kill-session request, the device returns a response in the following format if the kill-session operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

Configuration example

Configuration requirement

The user whose session's ID is 1 terminates the session with session ID 2.

Configuration procedure

# Enter XML view.

```
<Sysname> xml
```

# Exchange capabilities.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            urn:ietf:params:netconf:base:1.0
        </capability>
    </capabilities>
</hello>
```

# Terminate the session with session ID 2.

```
<rpc message-id="101"    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <kill-session>
       <session-id>2</session-id>
     </kill-session>
```

```
      </rpc>
```

Verifying the configuration

If the client receives the following text, the NETCONF session with session ID 2 has been terminated, and the client with session ID 2 has returned from XML view to user view:

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="101"  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
```

Returning to the CLI

To return from XML view to the CLI, send the following close-session request:

```
<?xml version="1.0"?>
   <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <close-session/>
   </rpc>
```

When the device receives the close-session request, it sends the following response and returns to CLI's user view:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <ok/>
</rpc-reply>
```

Appendix

Appendix A Supported NETCONF operations

Table 30 lists the NETCONF operations available with Comware V7.

Table 30 NETCONF operations

Operation	Description	XML example

get	Retrieves device configuration and state information.	To retrieve device configuration and state information for the Syslog module:
		<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:xc="http://www.hp.com/netconf/base:1.0">
		  <get>
		    <filter type="subtree">
		      <top xmlns="http://www.hp.com/netconf/data:1.0" >
		        <Syslog>
		        </Syslog>
		      </top>
		    </filter>
		  </get>
		</rpc>

get-config	Retrieves the non-default configuration data. If non-default configuration data does not exist, the device returns a response with empty data.	To retrieve non-default configuration data for the interface table:
		<rpc message-id ="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:xc="http://www.hp.com/netconf/base:1.0">
		  <get-config>
		    <source>
		      <running/>
		    </source>
		    <filter type="subtree">
		      <top xmlns="http://www.hp.com/netconf/config:1.0">
		        <Ifmgr>
		          <Interfaces>
		<Interface/>
		          </Interfaces>
		        </Ifmgr>
		      </top>
		    </filter>
		  </get-config>
		</rpc>

Operation	Description	XML example

get-bulk	Retrieves a number of data entries (including device configuration and state information) starting from the data entry next to the one with the specified index.	To retrieve device configuration and state information for all interface:
		`<rpc message-id ="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`  <get-bulk>`
		`    <filter type="subtree">`
		`      <top xmlns="http://www.hp.com/netconf/data:1.0">`
		`        <Ifmgr>`
		`          <Interfaces xc:count=”5” xmlns:xc=” http://www.hp.com/netconf/base:1.0”>`
		`                 <Interface/>`
		`          </Interfaces>`
		`        </Ifmgr>`
		`      </top>`
		`    </filter>`
		`  </get-bulk>`
		`</rpc>`

get-bulk-config	Retrieves a number of non-default configuration data entries starting from the data entry next to the one with the specified index.	To retrieve non-default configuration for all interfaces:
		`<rpc message-id ="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`  <get-bulk-config>`
		`    <source>`
		`      <running/>`
		`    </source>`
		`    <filter type="subtree">`
		`      <top xmlns="http://www.hp.com/netconf/config:1.0">`
		`        <Ifmgr>`
		`        </Ifmgr>`
		`      </top>`
		`    </filter>`
		`  </get-bulk-config>`
		`</rpc>`

Operation	Description	XML example

edit-config: merge	Changes the running configuration.
	To use the *merge* attribute in the edit-config operation, you must specify the operation target (on a specified level):
	• If the specified target exists, the operation directly changes the configuration for the target.
	• If the specified target does not exist, the operation creates and configures the target.
	• If the specified target does not exist and it cannot be created, an error message is returned.
		To change the buffer size to 120:
		`<rpc message-id ="101"`
		`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"`
		`xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`  <edit-config>`
		`    <target>`
		`      <running/>`
		`    </target>`
		`    <config>`
		`      <top`
		`xmlns="http://www.hp.com/netconf/config:1.0"><Syslog`
		`xmlns="http://www.hp.com/netconf/config:1.0" xc:operation="merge">`
		`    <LogBuffer>`
		`        <BufferSize>120</BufferSize>`
		`    </LogBuffer>`
		`</Syslog>`
		`      </top>`
		`    </config>`
		`  </edit-config>`
		`</rpc>`

edit-config: create	Creates a specified target. To use the *create* attribute in the edit-config operation, you must specify the operation target. • If the table supports target creation and the specified target does not exist, the operation creates and then configures the target. • If the specified target exists, a data-exist error message is returned.	The XML data format is the same as the edit-config message with the *merge* attribute. Change the operation attribute from **merge** to **create**.

Operation	Description	XML example
edit-config: replace	Replaces the specified target. - If the specified target exists, the operation replaces the configuration of the target with the configuration carried in the message. - If the specified target does not exist but is allowed to be created, create the target and then apply the configuration of the target. - If the specified target does not exist and is not allowed to be created, the operation is not conducted and an invalid-value error message is returned.	The syntax is the same as the edit-config message with the *merge* attribute. Change the operation attribute from **merge** to **replace**.
edit-config: remove	Removes the specified configuration. - If the specified target has only the table index, the operation removes all configuration of the specified target, and the target itself. - If the specified target has the table index and configuration data, the operation removes the specified configuration data of this target. - If the specified target does not exist, or the XML message does not specify any target, a success message is returned.	The syntax is the same as the edit-config message with the *merge* attribute. Change the operation attribute from **merge** to **remove**.

Operation	Description	XML example
edit-config: delete	Deletes the specified configuration. • If the specified target has only the table index, the operation removes all configuration of the specified target, and the target itself. • If the specified target has the table index and configuration data, the operation removes the specified configuration data of this target. • If the specified target does not exist, an error message is returned, showing that the target does not exist.	The syntax is the same as the edit-config message with the *merge* attribute. Change the operation attribute from **merge** to **delete**.

Operation	Description	XML example

edit-config: default-operation	Modifies the current configuration of the device using the default operation method.
	If you do not specify an operation attribute for an edit-config message, NETCONF uses one of the following default operation attributes: merge, create, delete, and replace. Your setting of the value for the <default-operation> element takes effect only once. If you do not specify an operation attribute and the default operation method for an <edit-config> message, **merge** is always applied.
	- **merge**—This is the default value for the <default-operation> element.
	- **replace**—This value is used when the operation attribute is not specified and the default operation method is specified as **replace**.
	- **none**—This value is used when the operation attribute is not specified and the default operation method is specified as **none**. If this value is specified, the edit-config operation is used only for schema verification rather than issuing a configuration. If the schema verification is passed, a successful message is returned. Otherwise, an error message is returned.
		To issue an empty operation for schema verification purposes:
		`<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`  <edit-config>`
		`    <target>`
		`      <running/>`
		`    </target>`
		`<default-operation>none</default-operation>`
		`    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`      <top xmlns="http://www.hp.com/netconf/config:1.0">`
		`        <Ifmgr >`
		`          <Interfaces>`
		`            <Interface>`
		`              <Index>262</Index>`
		`<Description>222222</Description>`
		`            </Interface>`
		`          </Interfaces>`
		`        </Ifmgr>`
		`      </top>`
		`    </config>`
		`  </edit-config>`
		`</rpc>`

Operation	Description	XML example

edit-config: error-option	Determines the action to take in case of a configuration error.
	The error-option element has one of the following values:
	• **stop-on-error**—Stops the operation on error and returns an error message. This is the default error-option value.
	• **continue-on-error**—Continues the operation on error and returns an error message.
	• **rollback-on-error**—Rolls back the configuration. This value is not supported by the current software version, and it is reserved for future use.
		To issue the configuration for two interfaces with the error-option element value as continue-on-error:
		`<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`<edit-config>`
		`<target>`
		`<running/>`
		`</target>`
		`<error-option>continue-on-error</error-option>`
		`<config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`<top xmlns="http://www.hp.com/netconf/config:1.0">`
		`<Ifmgr xc:operation="merge">`
		`<Interfaces>`
		`<Interface>`
		`<Index>262</Index>`
		`<Description>222</Description>`
		`<ConfigSpeed>100</ConfigSpeed>`
		`<ConfigDuplex>1</ConfigDuplex>`
		`</Interface>`
		`<Interface>`
		`<Index>263</Index>`
		`<Description>333</Description>`
		`<ConfigSpeed>100</ConfigSpeed>`
		`<ConfigDuplex>1</ConfigDuplex>`
		`</Interface>`
		`</Interfaces>`
		`</Ifmgr>`
		`</top>`
		`</config>`
		`</edit-config>`
		`</rpc>`

Operation	Description	XML example

edit-config: test-option	Determines whether to issue a configuration item in the edit-configure operation. The test-option element has one of the following values:
	• **test-then-set**—Performs a validation test before attempting to set. If the validation test fails, the edit-config operation is not performed. This is the default test-option value.
	• **set**—Directly performs the set operation without the validation test.
	• **test-only**—Performs only a validation test without attempting to set. If the validation test succeeds, a successful message is returned. Otherwise, an error message is returned.
		To issue the configuration for an interface for test purposes:
		`<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`  <edit-config>`
		`    <target>`
		`      <running/>`
		`</target>`
		`<test-option>test-only</test-option>`
		`    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`      <top xmlns="http://www.hp.com/netconf/config:1.0">`
		`        <Ifmgr xc:operation="merge">`
		`          <Interfaces>`
		`            <Interface>`
		`              <Index>262</Index>`
		`<Description>222</Description>`
		`<ConfigSpeed>100</ConfigSpeed>`
		`<ConfigDuplex>1</ConfigDuplex>`
		`               </Interface>`
		`          </Interfaces>`
		`        </Ifmgr>`
		`      </top>`
		`    </config>`
		`  </edit-config>`
		`</rpc>`

action	Issues actions that are not for configuring data, for example, reset action.	To clear statistics information for all interfaces:
		`<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`  <action>`
		`    <top xmlns="http://www.hp.com/netconf/action:1.0">`
		`      <Ifmgr>`
		`            <ClearAllIfStatistics>`
		`                <Clear>`
		`                </Clear>`
		`        </ClearAllIfStatistics>`
		`      </Ifmgr>`
		`    </top>`
		`  </action>`
		`</rpc>`

Operation	Description	XML example

lock	Locks the configuration data that can be changed by the edit-config operation. Other configurations are not limited by the lock operation.
	This lock operation locks only the configurations made through NETCONF sessions, rather than those through other protocols, for example, SNMP.
		To lock the configuration:
		`<rpc message-id="101"`
		`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		` <lock>`
		`    <target>`
		`        <running/>`
		`    </target>`
		`</lock>`
		`</rpc>`

unlock	Unlocks the configuration, so NETCONF sessions can change device configuration.
	When a NETCONF session is terminated, the related locked configuration is also unlocked.
		To unlock the configuration:
		`<rpc message-id="101"`
		`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`<unlock>`
		`    <target>`
		`        <running/>`
		`    </target>`
		`</unlock>`
		`</rpc>`

get-sessions	Retrieves information about all NETCONF sessions in the system.	To retrieve information about all NETCONF sessions in the system:
		`<rpc message-id="101"`
		`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`<get-sessions/>`
		`</rpc>`

close-session	Terminates the NETCONF session for the current user, to unlock the configuration and release the resources (for example, memory) of this session. This operation logs the current user off the XML view.	To terminate the NETCONF session for the current user:
		`<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`<close-session />`
		`</rpc>`

kill-session	Terminates the NETCONF session for another user. This operation cannot terminate the NETCONF session for the current user.	To terminate the NETCONF session with session-id 1:
		`<rpc message-id="101"`
		`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`<kill-session>`
		`      <session-id>1</session-id>`
		`  </kill-session>`
		`</rpc>`

Operation	Description	XML example

CLI	Executes CLI operations. A request message encloses commands in the <CLI> element, and a response message encloses the command output in the <CLI> element.
	NETCONF supports the following views:
	• **Execution**—User view.
	• **Configuration**—System view.
	To execute a command in other views, specify the command for entering the specified view, and then the desired command.
		To execute the **display this** command in system view:
		`<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`  <CLI>`
		`        <Configuration>display this</Configuration>`
		`  </CLI>`
		`</rpc>`

save	Saves the running configuration. You can use the <file> element to specify a file for saving the configuration. If the file column does not exist, the running configuration is automatically saved to the main next-startup configuration file.	To save the running configuration to the file **test.cfg**:
		`<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <save>`
		`    <file>test.cfg</file>`
		`  </save>`
		`</rpc>`

load	Loads the configuration. After the device finishes the load operation, the configuration in the specified file is merged into the current configuration of the device.	To merge the configuration in the file **a1.cfg** to the current configuration of the device:
		`<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <load>`
		`    <file>a1.cfg</file>`
		`  </load>`
		`</rpc>`

rollback	Rolls back the configuration. To do so, you must specify the configuration file in the <file> element. After the device finishes the rollback operation, the current device configuration is totally replaced with the configuration in the specified configuration file.	To roll back the current configuration to the configuration in the file **1A.cfg**:
		`<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">`
		`<rollback>`
		`    <file>1A.cfg</file>`
		`</rollback>`
		`</rpc>`

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 FlexNetwork Technology 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
- HP Education http://www.hp.com/learn

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.
Italic	*Italic* 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.

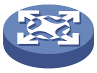
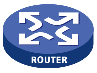
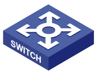
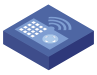
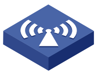
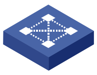
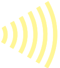
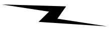
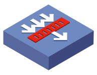
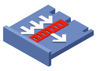
GUI conventions

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.
	Represents an access controller, a unified wired-WLAN module, or the switching engine on a unified wired-WLAN switch.
	Represents an access point.
	Represents a mesh access point.
	Represents omnidirectional signals.
	Represents directional signals.
	Represents a security product, such as a firewall, UTM, multiservice security gateway, or load-balancing device.
	Represents a security card, such as a firewall, load-balancing, NetStream, SSL VPN, IPS, or ACG card.

Port numbering in examples

The port numbers in this document are for illustration only and might be unavailable on your device.

Index