Copyright © 2016-2018, New H3C Technologies Co., Ltd. and its licensors

All rights reserved

No part of this manual may be reproduced or transmitted in any form or by any means without prior written consent of New H3C Technologies Co., Ltd.

Trademarks

H3C, H3CS, H3CIE, H3CNE, Aolynk, H³Care, IRF, NetPilot, Netflow, SecEngine, SecPath, SecCenter, SecBlade, Comware, ITCMM and HUASAN are trademarks of New H3C Technologies Co., Ltd.

All other trademarks that may be mentioned in this manual are the property of their respective owners

Notice

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute the warranty of any kind, express or implied.
Preface

This configuration guide describes network management and monitoring fundamentals and configuration procedures for the H3C S5130-EI switch series. It describes how to view system information, collect traffic statistics, assess the network performance, synchronize time for all devices with clocks in your network, and use the **ping**, **tracert**, and **debug** commands to check and debug the current network connectivity.

This preface includes the following topics about the documentation:

- **Audience**
- **Conventions**
- **Documentation feedback**

Audience

This documentation is intended for:

- Network planners.
- Field technical support and servicing engineers.
- Network administrators working with the S5130-EI switch series.

Conventions

The following information describes the conventions used in the documentation.

**Command conventions**

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Boldface</strong></td>
<td><strong>Bold</strong> text represents commands and keywords that you enter literally as shown.</td>
</tr>
<tr>
<td><strong>Italic</strong></td>
<td><strong>Italic</strong> text represents arguments that you replace with actual values.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Square brackets enclose syntax choices (keywords or arguments) that are optional.</td>
</tr>
<tr>
<td>{ x</td>
<td>y</td>
</tr>
<tr>
<td>[ x</td>
<td>y</td>
</tr>
<tr>
<td>{ x</td>
<td>y</td>
</tr>
<tr>
<td>[ x</td>
<td>y</td>
</tr>
<tr>
<td>&amp;&lt;1-n&gt;</td>
<td>The argument or keyword and argument combination before the ampersand (&amp;) sign can be entered 1 to n times.</td>
</tr>
<tr>
<td>#</td>
<td>A line that starts with a pound (#) sign is comments.</td>
</tr>
</tbody>
</table>

**GUI conventions**

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Boldface</strong></td>
<td>Window names, button names, field names, and menu items are in Boldface. For example, the <strong>New User</strong> window opens; click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>
Multi-level menus are separated by angle brackets. For example, **File > Create > Folder**.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WARNING!</td>
<td>An alert that calls attention to important information that if not understood or followed can result in personal injury.</td>
</tr>
<tr>
<td>CAUTION:</td>
<td>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.</td>
</tr>
<tr>
<td>IMPORTANT:</td>
<td>An alert that calls attention to essential information.</td>
</tr>
<tr>
<td>NOTE:</td>
<td>An alert that contains additional or supplementary information.</td>
</tr>
<tr>
<td>TIP:</td>
<td>An alert that provides helpful information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network topology icons</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Represents a generic network device, such as a router, switch, or firewall.</td>
<td></td>
</tr>
<tr>
<td>Represents a routing-capable device, such as a router or Layer 3 switch.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Represents an access controller, a unified wired-WLAN module, or the access controller engine on a unified wired-WLAN switch.</td>
<td></td>
</tr>
<tr>
<td>Represents an access point.</td>
<td></td>
</tr>
<tr>
<td>Represents a wireless terminator unit.</td>
<td></td>
</tr>
<tr>
<td>Represents a wireless terminator.</td>
<td></td>
</tr>
<tr>
<td>Represents a mesh access point.</td>
<td></td>
</tr>
<tr>
<td>Represents omnidirectional signals.</td>
<td></td>
</tr>
<tr>
<td>Represents directional signals.</td>
<td></td>
</tr>
<tr>
<td>Represents a security product, such as a firewall, UTM, multiservice security gateway, or load balancing device.</td>
<td></td>
</tr>
<tr>
<td>Represents a security module, such as a firewall, load balancing, NetStream, SSL VPN, IPS, or ACG module.</td>
<td></td>
</tr>
</tbody>
</table>
Examples provided in this document

Examples in this document might use devices that differ from your device in hardware model, configuration, or software version. It is normal that the port numbers, sample output, screenshots, and other information in the examples differ from what you have on your device.

Documentation feedback

You can e-mail your comments about product documentation to info@h3c.com.
We appreciate your comments.
# Contents

## Using ping, tracert, and system debugging

- Ping
  - Using a ping command to test network connectivity
  - Ping example
- Tracert
  - Prerequisites
  - Using a tracert command to identify failed or all nodes in a path
  - Tracert example
- System debugging
  - Debugging information control switches
  - Debugging a feature module

## Configuring NTP

- Overview
  - How NTP works
  - NTP architecture
  - Association modes
  - NTP security
  - Protocols and standards
- Configuration restrictions and guidelines
- Configuration task list
- Enabling the NTP service
- Configuring NTP association modes
  - Configuring NTP in client/server mode
  - Configuring NTP in symmetric active/passive mode
  - Configuring NTP in broadcast mode
  - Configuring NTP in multicast mode
- Configuring access control rights
- Configuring NTP authentication
  - Configuring NTP authentication in client/server mode
  - Configuring NTP authentication in symmetric active/passive mode
  - Configuring NTP authentication in broadcast mode
  - Configuring NTP authentication in multicast mode
- Configuring NTP optional parameters
  - Specifying the source interface for NTP messages
  - Disabling an interface from receiving NTP messages
  - Configuring the maximum number of dynamic associations
  - Configuring a DSCP value for NTP packets
- Configuring the local clock as a reference source
- Displaying and maintaining NTP

## Configuring SNTP

- Configuration restrictions and guidelines
- Configuration task list
- Enabling the SNTP service
- Specifying an NTP server for the device
- Configuring SNTP authentication
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring kernel threads</td>
<td>173</td>
</tr>
<tr>
<td>Configuring kernel thread deadloop detection</td>
<td>173</td>
</tr>
<tr>
<td>Configuring kernel thread starvation detection</td>
<td>174</td>
</tr>
<tr>
<td>Displaying and maintaining kernel threads</td>
<td>174</td>
</tr>
<tr>
<td><strong>Configuring EAA</strong></td>
<td>176</td>
</tr>
<tr>
<td>Overview</td>
<td>176</td>
</tr>
<tr>
<td>EAA framework</td>
<td>176</td>
</tr>
<tr>
<td>Elements in a monitor policy</td>
<td>177</td>
</tr>
<tr>
<td>EAA environment variables</td>
<td>178</td>
</tr>
<tr>
<td>Configuring a user-defined EAA environment variable</td>
<td>179</td>
</tr>
<tr>
<td>Configuring a monitor policy</td>
<td>179</td>
</tr>
<tr>
<td>Configuration restrictions and guidelines</td>
<td>179</td>
</tr>
<tr>
<td>Configuring a monitor policy from the CLI</td>
<td>180</td>
</tr>
<tr>
<td>Configuring a monitor policy by using Tcl</td>
<td>181</td>
</tr>
<tr>
<td>Suspending monitor policies</td>
<td>182</td>
</tr>
<tr>
<td>Displaying and maintaining EAA settings</td>
<td>182</td>
</tr>
<tr>
<td>Configuration examples</td>
<td>183</td>
</tr>
<tr>
<td>CLI-defined policy configuration example</td>
<td>183</td>
</tr>
<tr>
<td>CLI-defined policy with EAA environment variables configuration example</td>
<td>184</td>
</tr>
<tr>
<td>Tcl-defined policy configuration example</td>
<td>185</td>
</tr>
<tr>
<td><strong>Configuring CWMP</strong></td>
<td>187</td>
</tr>
<tr>
<td>Overview</td>
<td>187</td>
</tr>
<tr>
<td>CWMP network framework</td>
<td>187</td>
</tr>
<tr>
<td>Basic CWMP functions</td>
<td>187</td>
</tr>
<tr>
<td>How CWMP works</td>
<td>188</td>
</tr>
<tr>
<td>Configuration task list</td>
<td>190</td>
</tr>
<tr>
<td>Enabling CWMP from the CLI</td>
<td>191</td>
</tr>
<tr>
<td>Configuring ACS attributes</td>
<td>191</td>
</tr>
<tr>
<td>Configuring the preferred ACS attributes</td>
<td>191</td>
</tr>
<tr>
<td>Configuring the default ACS attributes from the CLI</td>
<td>193</td>
</tr>
<tr>
<td>Configuring CPE attributes</td>
<td>193</td>
</tr>
<tr>
<td>Configuring ACS authentication parameters</td>
<td>193</td>
</tr>
<tr>
<td>Configuring the provision code</td>
<td>193</td>
</tr>
<tr>
<td>Configuring the CWMP connection interface</td>
<td>194</td>
</tr>
<tr>
<td>Configuring autoconnect parameters</td>
<td>194</td>
</tr>
<tr>
<td>Enabling NAT traversal for the CPE</td>
<td>195</td>
</tr>
<tr>
<td>Specifying an SSL client policy for HTTPS connection to ACS</td>
<td>196</td>
</tr>
<tr>
<td>Displaying and maintaining CWMP</td>
<td>196</td>
</tr>
<tr>
<td>CWMP configuration example</td>
<td>196</td>
</tr>
<tr>
<td>Network requirements</td>
<td>196</td>
</tr>
<tr>
<td>Configuration procedure</td>
<td>197</td>
</tr>
<tr>
<td>Verifying the configuration</td>
<td>206</td>
</tr>
<tr>
<td><strong>Configuring PoE</strong></td>
<td>207</td>
</tr>
<tr>
<td>Overview</td>
<td>207</td>
</tr>
<tr>
<td>PoE configuration task list</td>
<td>207</td>
</tr>
<tr>
<td>Enabling PoE for a PI</td>
<td>208</td>
</tr>
<tr>
<td>Enabling nonstandard PD detection</td>
<td>209</td>
</tr>
<tr>
<td>Configuring the PD detection mode</td>
<td>209</td>
</tr>
<tr>
<td>Configuring the maximum PI power</td>
<td>209</td>
</tr>
<tr>
<td>Configuring PI power management</td>
<td>210</td>
</tr>
<tr>
<td>Configuring PoE monitoring</td>
<td>211</td>
</tr>
<tr>
<td>Configuring PSE power monitoring</td>
<td>211</td>
</tr>
<tr>
<td>Configuring a PI by using a PoE profile</td>
<td>211</td>
</tr>
<tr>
<td>Configuring a PoE profile</td>
<td>212</td>
</tr>
<tr>
<td>Applying a PoE profile</td>
<td>212</td>
</tr>
<tr>
<td>Upgrading PSE firmware in service</td>
<td>212</td>
</tr>
<tr>
<td>Displaying and maintaining PoE</td>
<td>213</td>
</tr>
<tr>
<td>PoE configuration example</td>
<td>213</td>
</tr>
<tr>
<td>Troubleshooting PoE</td>
<td>214</td>
</tr>
</tbody>
</table>
# Configuring NETCONF

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>216</td>
</tr>
<tr>
<td>NETCONF structure</td>
<td>216</td>
</tr>
<tr>
<td>NETCONF message format</td>
<td>217</td>
</tr>
<tr>
<td>How to use NETCONF</td>
<td>218</td>
</tr>
<tr>
<td>Protocols and standards</td>
<td>218</td>
</tr>
<tr>
<td>FIPS compliance</td>
<td>219</td>
</tr>
<tr>
<td>NETCONF configuration task list</td>
<td>219</td>
</tr>
<tr>
<td>Enabling NETCONF over SOAP</td>
<td>219</td>
</tr>
<tr>
<td>Enabling NETCONF over SSH</td>
<td>220</td>
</tr>
<tr>
<td>Enabling NETCONF logging</td>
<td>220</td>
</tr>
<tr>
<td>Establishing a NETCONF session</td>
<td>220</td>
</tr>
<tr>
<td>Entering XML view</td>
<td>220</td>
</tr>
<tr>
<td>Exchanging capabilities</td>
<td>221</td>
</tr>
<tr>
<td>Subscribing to event notifications</td>
<td>221</td>
</tr>
<tr>
<td>Subscription procedure</td>
<td>221</td>
</tr>
<tr>
<td>Example for subscribing to event notifications</td>
<td>222</td>
</tr>
<tr>
<td>Locking/unlocking the configuration</td>
<td>223</td>
</tr>
<tr>
<td>Locking the configuration</td>
<td>224</td>
</tr>
<tr>
<td>Unlocking the configuration</td>
<td>224</td>
</tr>
<tr>
<td>Example for locking the configuration</td>
<td>224</td>
</tr>
<tr>
<td>Performing service operations</td>
<td>225</td>
</tr>
<tr>
<td>Performing the get/get-bulk operation</td>
<td>226</td>
</tr>
<tr>
<td>Performing the get-config/get-bulk-config operation</td>
<td>227</td>
</tr>
<tr>
<td>Performing the edit-config operation</td>
<td>228</td>
</tr>
<tr>
<td>All-module configuration data retrieval example</td>
<td>228</td>
</tr>
<tr>
<td>Syslog configuration data retrieval example</td>
<td>230</td>
</tr>
<tr>
<td>Example for retrieving a data entry for the interface table</td>
<td>231</td>
</tr>
<tr>
<td>Example for changing the value of a parameter</td>
<td>232</td>
</tr>
<tr>
<td>Saving, rolling back, and loading the configuration</td>
<td>233</td>
</tr>
<tr>
<td>Saving the configuration</td>
<td>233</td>
</tr>
<tr>
<td>Rolling back the configuration based on a configuration file</td>
<td>233</td>
</tr>
<tr>
<td>Rolling back the configuration based on a rollback point</td>
<td>234</td>
</tr>
<tr>
<td>Loading the configuration</td>
<td>238</td>
</tr>
<tr>
<td>Example for saving the configuration</td>
<td>238</td>
</tr>
<tr>
<td>Filtering data</td>
<td>239</td>
</tr>
<tr>
<td>Table-based filtering</td>
<td>239</td>
</tr>
<tr>
<td>Column-based filtering</td>
<td>239</td>
</tr>
<tr>
<td>Example for filtering data with regular expression match</td>
<td>242</td>
</tr>
<tr>
<td>Example for filtering data by conditional match</td>
<td>243</td>
</tr>
<tr>
<td>Performing CLI operations through NETCONF</td>
<td>244</td>
</tr>
<tr>
<td>Configuration procedure</td>
<td>244</td>
</tr>
<tr>
<td>CLI operation example</td>
<td>245</td>
</tr>
<tr>
<td>Retrieving NETCONF session information</td>
<td>246</td>
</tr>
<tr>
<td>Terminating another NETCONF session</td>
<td>247</td>
</tr>
<tr>
<td>Configuration procedure</td>
<td>247</td>
</tr>
<tr>
<td>Configuration example</td>
<td>247</td>
</tr>
<tr>
<td>Returning to the CLI</td>
<td>248</td>
</tr>
<tr>
<td>Configuring the packet capture</td>
<td>249</td>
</tr>
<tr>
<td>Overview</td>
<td>249</td>
</tr>
<tr>
<td>Filter elements</td>
<td>249</td>
</tr>
<tr>
<td>Building a capture filter</td>
<td>254</td>
</tr>
<tr>
<td>Building a display filter</td>
<td>255</td>
</tr>
<tr>
<td>Configuring packet capture</td>
<td>256</td>
</tr>
<tr>
<td>Saving captured packets to a file</td>
<td>256</td>
</tr>
<tr>
<td>Filtering packet data to display</td>
<td>256</td>
</tr>
<tr>
<td>Displaying the contents in a packet file</td>
<td>257</td>
</tr>
<tr>
<td>Packet capture configuration examples</td>
<td>257</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine if a specified address in an IP network is reachable.</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>• For IPv6 networks: ping ipv6 [-a source-ipv6</td>
</tr>
</tbody>
</table>

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.
Configuration procedure

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

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 that:
- Device A has sent five ICMP packets to Device C and Device A has received 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 from Device A to 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) enables retrieval of 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 H3C 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 H3C 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 H3C 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 H3C 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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
</table>
| 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*

Device A 1.1.1.1/24 1.1.1.2/24 Device B 1.1.2.1/24 1.1.2.2/24 Device C

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. To identify the cause of the problem, execute the following commands on Device A and Device C:
   - Execute the `debugging ip icmp` command and verify that Device A and Device C can send and receive the correct ICMP packets.
   - Execute the `display ip routing-table` command to verify that Device A and Device C have a route to each other.

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, 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](image)

**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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enable debugging for a</td>
<td><code>debugging module-name [option]</code></td>
<td>By default, all debugging functions are disabled.</td>
</tr>
<tr>
<td>module in user view.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. (Optional.) Display the</td>
<td><code>display debugging [module-name]</code></td>
<td>N/A</td>
</tr>
<tr>
<td>enabled debugging in any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>view.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 of Device A is set to 10:00:00 am and that of Device B is set to 11:00:00 am.
- 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 strata 1 to 16 to define clock accuracy, as shown in Figure 6. A lower stratum value represents higher accuracy. Clocks at strata 1 through 15 are in synchronized state, and clocks at stratum 16 are not synchronized.

*Figure 6 NTP architecture*

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

<table>
<thead>
<tr>
<th>Mode</th>
<th>Working process</th>
<th>Principle</th>
<th>Application scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client/server</td>
<td>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.</td>
<td>A client can be synchronized to a server, but a server cannot be synchronized to a client.</td>
<td>As Figure 6 shows, this mode is intended for configurations where devices of a higher stratum are synchronized to devices with a lower stratum.</td>
</tr>
<tr>
<td>Symmetric active/passive</td>
<td>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.</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>Mode</td>
<td>Working process</td>
<td>Principle</td>
<td>Application scenario</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Broadcast</td>
<td>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.</td>
<td>A broadcast client can be synchronized to a broadcast server, but a broadcast server cannot be synchronized to a broadcast client.</td>
<td>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.</td>
</tr>
<tr>
<td>Multicast</td>
<td>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.</td>
<td>A multicast client can be synchronized to a multicast server, but a multicast server cannot be synchronized to a multicast client.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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**


**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.
- As a best practice to ensure time synchronization accuracy, do not specify 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

<table>
<thead>
<tr>
<th>Tasks at a glance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Required.) Enabling the NTP service</td>
</tr>
<tr>
<td>(Required.) Perform at least one of the following tasks:</td>
</tr>
<tr>
<td>• Configuring NTP association modes</td>
</tr>
<tr>
<td>• Configuring the local clock as a reference source</td>
</tr>
<tr>
<td>(Optional.) Configuring access control rights</td>
</tr>
<tr>
<td>(Optional.) Configuring NTP authentication</td>
</tr>
<tr>
<td>(Optional.) Configuring NTP optional parameters</td>
</tr>
</tbody>
</table>

Enabling the NTP service

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>ntp-service enable</td>
<td>By default, the NTP service is not enabled.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>ntp-service unicast-peer { peer-name</td>
<td>ip-address } [ authentication-keyid keyid</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td><strong>system-view</strong></td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td><strong>interface interface-type interface-number</strong></td>
</tr>
<tr>
<td>3.</td>
<td>Configure the device to operate in broadcast client mode.</td>
<td><strong>ntp-service broadcast-client</strong></td>
</tr>
</tbody>
</table>

Configuring the broadcast server

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td><strong>system-view</strong></td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td><strong>interface interface-type interface-number</strong></td>
</tr>
<tr>
<td>3.</td>
<td>Configure the device to operate in NTP broadcast server mode.</td>
<td>**ntp-service broadcast-server [ authentication-keyid keyid</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td><strong>system-view</strong></td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td><strong>interface interface-type interface-number</strong></td>
</tr>
</tbody>
</table>
### Configuring the multicast client

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td></td>
<td>Configure the device to operate in multicast client mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configure the device to operate in multicast client mode: <code>ntp-service multicast-client [ ip-address ]</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configure the device to operate in IPv6 multicast client mode: <code>ntp-service ipv6 multicast-client ipv6-multicast-address</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

### Configuring the multicast server

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td><code>system-view</code></td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td><code>interface interface-type interface-number</code></td>
</tr>
<tr>
<td>3.</td>
<td>Configure the device to operate in multicast server mode.</td>
<td>- Configure the device to operate in multicast server mode: <code>ntp-service multicast-server [ ip-address ] [ authentication-keyid keyid ] [ ttl ttl-number ] [ version number ]</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configure the device to operate in IPv6 multicast server mode: <code>ntp-service ipv6 multicast-server ipv6-multicast-address [ authentication-keyid keyid ] [ ttl ttl-number ]</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

### Configuring access control rights

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td><code>system-view</code></td>
</tr>
</tbody>
</table>
**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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable NTP authentication.</td>
<td>ntp-service authentication enable</td>
</tr>
<tr>
<td>3.</td>
<td>Configure an NTP authentication key.</td>
<td>ntp-service authentication-keyid keyid authentication-mode md5 cipher simple value</td>
</tr>
<tr>
<td>4.</td>
<td>Configure the key as a trusted key.</td>
<td>ntp-service reliable authentication-keyid keyid</td>
</tr>
</tbody>
</table>
Step | Command | Remarks
--- | --- | ---
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enable NTP authentication.</td>
<td><code>ntp-service authentication enable</code></td>
<td>By default, NTP authentication is disabled.</td>
</tr>
<tr>
<td>3. Configure an NTP authentication key.</td>
<td>`ntp-service authentication-keyid keyid authentication-mode md5 { cipher</td>
<td>simple } value`</td>
</tr>
<tr>
<td>4. Configure the key as a trusted key.</td>
<td><code>ntp-service reliable authentication-keyid keyid</code></td>
<td>By default, no authentication key is configured as a trusted key.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
<th>Authentication result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NTP authentication</td>
<td>Configure a key and configure it as a trusted key</td>
<td>Associate the key with an NTP server</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>ntp-service authentication enable</td>
<td>By default, NTP authentication is disabled.</td>
</tr>
<tr>
<td>3.</td>
<td>ntp-service authentication-keyid keyid authentication-mode md5 { cipher</td>
<td>simple } value</td>
</tr>
<tr>
<td>4.</td>
<td>ntp-service reliable authentication-keyid keyid</td>
<td>By default, no authentication key is configured as a trusted key.</td>
</tr>
</tbody>
</table>
| 5.   | • 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
2. | Enable NTP authentication. | ntp-service authentication enable
3. | Configure an NTP authentication key. | ntp-service authentication-keyid keyid authentication-mode md5 { cipher | simple } value
4. | Configure the key as a trusted key. | ntp-service reliable authentication-keyid keyid

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

<table>
<thead>
<tr>
<th>Active peer</th>
<th>Passive peer</th>
<th>Authentication result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enable NTP authentication</td>
<td>Configure a key and configure it as a trusted key</td>
<td>Associate the key with a passive peer</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Stratum level of the active and passive peers is not considered.

The active peer has a higher stratum than the passive peer.
<table>
<thead>
<tr>
<th>Active peer</th>
<th>Passive peer</th>
<th>Authentication result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NTP authentication</td>
<td>Configure a key and configure it as a trusted key</td>
<td>Associate the key with a passive peer</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>The passive peer has a higher stratum than the active peer.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable NTP authentication.</td>
<td>ntp-service authentication enable</td>
</tr>
<tr>
<td>3.</td>
<td>Configure an NTP authentication key.</td>
<td>ntp-service authentication-keyid keyid authentication-mode md5 { cipher</td>
</tr>
<tr>
<td>4.</td>
<td>Configure the key as a trusted key.</td>
<td>ntp-service reliable authentication-keyid keyid</td>
</tr>
</tbody>
</table>

To configure NTP authentication for a broadcast server:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable NTP authentication.</td>
<td>ntp-service authentication enable</td>
</tr>
</tbody>
</table>
3. Configure an NTP authentication key.
   - **Command**: `ntp-service authentication-keyid keyid authentication-mode md5 { cipher | simple } value`
   - **Remarks**: By default, no NTP authentication key is configured.

4. Configure the key as a trusted key.
   - **Command**: `ntp-service reliable authentication-keyid keyid`
   - **Remarks**: By default, no authentication key is configured as a trusted key.

5. Enter interface view.
   - **Command**: `interface interface-type interface-number`
   - **Remarks**: N/A

6. Associate the specified key with the broadcast server.
   - **Command**: `ntp-service broadcast-server authentication-keyid keyid`
   - **Remarks**: 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

<table>
<thead>
<tr>
<th>Enable NTP authentication</th>
<th>Configure a key and configure it as a trusted key</th>
<th>Associate the key with a broadcast server</th>
<th>Enable NTP authentication</th>
<th>Configure a key and configure it as a trusted key</th>
<th>Authentication result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Succeeded. NTP messages can be sent and received correctly.</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Failed. NTP messages cannot be sent and received correctly.</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>Failed. NTP messages cannot be sent and received correctly.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>Failed. NTP messages cannot be sent and received correctly.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>No authentication. NTP messages can be sent and received correctly.</td>
</tr>
<tr>
<td>Yes</td>
<td>N/A</td>
<td>No</td>
<td>Yes</td>
<td>N/A</td>
<td>Failed. NTP messages cannot be sent and received correctly.</td>
</tr>
<tr>
<td>Yes</td>
<td>N/A</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
<td>No authentication. NTP messages can be sent and received correctly.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable NTP authentication.</td>
<td>ntp-service authentication enable</td>
</tr>
<tr>
<td>3.</td>
<td>Configure an NTP authentication key.</td>
<td>ntp-service authentication-keyid keyid authentication-mode md5 { cipher</td>
</tr>
<tr>
<td>4.</td>
<td>Configure the key as a trusted key.</td>
<td>ntp-service reliable authentication-keyid keyid</td>
</tr>
</tbody>
</table>

To configure NTP authentication for a multicast server:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable NTP authentication.</td>
<td>ntp-service authentication enable</td>
</tr>
<tr>
<td>3.</td>
<td>Configure an NTP authentication key.</td>
<td>ntp-service authentication-keyid keyid authentication-mode md5 { cipher</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 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**

<table>
<thead>
<tr>
<th>Multicast server</th>
<th>Multicast client</th>
<th>Authentication result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable NTP authentication</td>
<td>Configure a key and configure it as a trusted key</td>
<td>Associate the key with a multicast server</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>N/A</td>
<td>No</td>
</tr>
</tbody>
</table>
### 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 acts 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specify the source interface for NTP messages.</td>
<td>By default, no source interface is specified for NTP messages.</td>
</tr>
<tr>
<td></td>
<td>• Specify the source interface for NTP messages: <code>ntp-service source interface-type interface-number</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Specify the source interface for IPv6 NTP messages: <code>ntp-service ipv6 source interface-type interface-number</code></td>
<td></td>
</tr>
</tbody>
</table>

Disabling an interface from receiving 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 receiving NTP messages.

To disable an interface from receiving NTP messages:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td><code>system-view</code></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td><code>interface interface-type interface-number</code></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Disable the interface from receiving IPv4 NTP messages.</td>
<td>By default, an interface receives IPv4 NTP messages.</td>
</tr>
<tr>
<td></td>
<td><code>undo ntp-service inbound enable</code></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Disable the interface from receiving IPv6 NTP messages.</td>
<td>By default, an interface receives IPv6 NTP messages.</td>
</tr>
<tr>
<td></td>
<td><code>undo ntp-service ipv6 inbound enable</code></td>
<td></td>
</tr>
</tbody>
</table>

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:
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
</tbody>
</table>
| 2.   | set a DSCP value for NTP packets. | 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>ntp-service refclock-master [ ip-address ] [ stratum ]</td>
<td>By default, the device does not use the local clock as a reference source.</td>
</tr>
</tbody>
</table>

Displaying and maintaining NTP

Execute **display** commands in any view.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about IPv6 NTP associations.</td>
<td>display ntp-service ipv6 sessions [ verbose ]</td>
</tr>
<tr>
<td>Display information about IPv4 NTP associations.</td>
<td>display ntp-service sessions [ verbose ]</td>
</tr>
<tr>
<td>Display information about NTP service status.</td>
<td>display ntp-service status</td>
</tr>
</tbody>
</table>
NTP configuration examples

NTP client/server mode configuration example

Network requirements

As shown in Figure 8, perform the following tasks:

- 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
IPv6 NTP client/server mode configuration example

Network requirements
As shown in Figure 9, perform the following tasks:

- 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

![Network Diagram](image)

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.
NTP symmetric active/passive mode configuration example

Network requirements

As shown in Figure 10, perform the following tasks:

- 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

```
Symmetric active peer Symmetric passive peer

30.1.3.24 30.1.3.224
Device A Device B
```

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, perform the following tasks:

- 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

Symmetric active peer   Symmetric passive peer

[Diagram showing Device A and Device B with IPv6 addresses 3000::35/64 and 3000::36/64]
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              Clock 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**

![Network diagram showing Switch C as NTP broadcast server, Switch A and Switch B as broadcast clients.]  

**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.
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:
   o Receive the multicast messages from Switch C without being enabled with the multicast functions.
   o 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 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 gigabitethernet 1/0/1
[SwitchB-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 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 gigabitethernet 1/0/1
   [SwitchB-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, perform the following tasks:
  • 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

![Network diagram](image)

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**

![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.

```bash
[SwitchB] interface vlan-interface 2
[SwitchB-Vlan-interface2] ntp-service broadcast-client
```

4. **Configure Switch C:**

   # Enable the NTP service.
   ```bash
   <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.

   ```bash
   [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.

   ```bash
   [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.

   ```bash
   [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

<table>
<thead>
<tr>
<th>source</th>
<th>reference</th>
<th>stra</th>
<th>reach</th>
<th>poll</th>
<th>now</th>
<th>offset</th>
<th>delay</th>
<th>disper</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0.1.31</td>
<td>127.127.1.0</td>
<td>3</td>
<td>3</td>
<td>64</td>
<td>68</td>
<td>-0.0</td>
<td>0.0000</td>
<td>0.0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Tasks at a glance</th>
<th>(Required.)</th>
<th>Enabling the SNTP service</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Required.)</td>
<td>Specifying an NTP server for the device</td>
</tr>
<tr>
<td></td>
<td>(Optional.)</td>
<td>Configuring SNTP authentication</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enable the SNTP service.</td>
<td>sntp enable</td>
<td>By default, the SNTP service is not enabled.</td>
</tr>
</tbody>
</table>

Specifying an NTP server for the device

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>sntp authentication enable</td>
<td>By default, SNTP authentication is disabled.</td>
</tr>
<tr>
<td>3.</td>
<td>sntp authentication-keyid keyid authentication-mode md5 { cipher</td>
<td>simple } value</td>
</tr>
<tr>
<td>4.</td>
<td>sntp reliable authentication-keyid keyid</td>
<td>By default, no trusted key is specified.</td>
</tr>
</tbody>
</table>
### Displaying and maintaining SNTP

Execute `display` commands in any view.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about all IPv6 SNTP associations.</td>
<td><code>display sntp ipv6 sessions</code></td>
</tr>
<tr>
<td>Display information about all IPv4 SNTP associations.</td>
<td><code>display sntp sessions</code></td>
</tr>
</tbody>
</table>

### SNTP configuration example

#### Network requirements

As shown in **Figure 17**, perform the following tasks:

- 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

<table>
<thead>
<tr>
<th>NTP server</th>
<th>Stratum</th>
<th>Version</th>
<th>Last receive time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.1.11</td>
<td>2</td>
<td>4</td>
<td>Tue, May 17 2011 9:11:20.833 (Synced)</td>
</tr>
</tbody>
</table>
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 18 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

<table>
<thead>
<tr>
<th>Severity value</th>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Emergency</td>
<td>The system is unusable. For example, the system authorization has expired.</td>
</tr>
<tr>
<td>1</td>
<td>Alert</td>
<td>Action must be taken immediately. For example, traffic on an interface exceeds the upper limit.</td>
</tr>
<tr>
<td>2</td>
<td>Critical</td>
<td>Critical condition. For example, the device temperature exceeds the upper limit, the power module fails, or the fan tray fails.</td>
</tr>
</tbody>
</table>
### Severit y value

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Error condition. For example, the link state changes.</td>
</tr>
<tr>
<td>4</td>
<td>Warning condition. For example, an interface is disconnected, or the memory resources are used up.</td>
</tr>
<tr>
<td>5</td>
<td>Notification Normal but significant condition. For example, a terminal logs in to the device, or the device reboots.</td>
</tr>
<tr>
<td>6</td>
<td>Informational Information message. For example, a command or a ping operation is executed.</td>
</tr>
<tr>
<td>7</td>
<td>Debugging Debug message.</td>
</tr>
</tbody>
</table>

### 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**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Log source modules</th>
<th>Output switch</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Debug</td>
</tr>
<tr>
<td>Monitor terminal</td>
<td>All supported modules</td>
<td>Disabled</td>
<td>Debug</td>
</tr>
<tr>
<td>Log host</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Informational</td>
</tr>
<tr>
<td>Log buffer</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Informational</td>
</tr>
<tr>
<td>Log file</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Informational</td>
</tr>
</tbody>
</table>

### 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**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Log source modules</th>
<th>Output switch</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnostic log file</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Debug</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Log source modules</th>
<th>Output switch</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security log file</td>
<td>All supported modules</td>
<td>Disabled</td>
<td>Debugging</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Log source modules</th>
<th>Output switch</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log host</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Informational</td>
</tr>
<tr>
<td>Log buffer</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Informational</td>
</tr>
<tr>
<td>Log file</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Informational</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Log source modules</th>
<th>Output switch</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace log file</td>
<td>All supported modules</td>
<td>Enabled</td>
<td>Debugging</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Output destination</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console, monitor terminal, log buffer, or</td>
<td>Prefix Timestamp Sysname</td>
<td>%Nov 24 14:21:43:502 2010</td>
</tr>
<tr>
<td>log file</td>
<td>Module/Level/Mnemonic: Content</td>
<td>Sysname SYSLOG/6/SYSLOG_RESTART: System restarted –</td>
</tr>
<tr>
<td></td>
<td></td>
<td>H3C Comware Software.</td>
</tr>
</tbody>
</table>
### Table 13 Log field description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix (information type)</td>
<td>A log to a destination other than the log host has an identifier in front of the timestamp:</td>
</tr>
<tr>
<td></td>
<td>1. An identifier of percent sign (%) indicates a log with a level equal to or higher than informational.</td>
</tr>
<tr>
<td></td>
<td>2. An identifier of asterisk (*) indicates a debug log or a trace log.</td>
</tr>
<tr>
<td></td>
<td>3. An identifier of caret (^) indicates a diagnostic log.</td>
</tr>
<tr>
<td>PRI (priority)</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>1. 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.</td>
</tr>
<tr>
<td></td>
<td>2. level ranges from 0 to 7. See Table 6 for more information about severity levels.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Records the time when the log was generated.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Hostip</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>This field exists only in logs in unicom format that are sent to the log host.</td>
</tr>
<tr>
<td>Serial number</td>
<td>Serial number of the device that generated the log.</td>
</tr>
<tr>
<td></td>
<td>This field exists only in logs in unicom format that are sent to the log host.</td>
</tr>
<tr>
<td>Sysname (host name or host IP address)</td>
<td>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.</td>
</tr>
<tr>
<td>%% (vendor ID)</td>
<td>Indicates that the information was generated by an H3C device.</td>
</tr>
<tr>
<td></td>
<td>This field exists only in logs sent to the log host.</td>
</tr>
</tbody>
</table>
### Field Description

**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

<table>
<thead>
<tr>
<th>Item</th>
<th>Destined to the log host</th>
<th>Destined to the console, monitor terminal, log buffer, and log file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Seconds</td>
<td>Milliseconds</td>
</tr>
<tr>
<td>Command used</td>
<td>info-center timestamp loghost</td>
<td>info-center timestamp</td>
</tr>
<tr>
<td>to set the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>timestamp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>format</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 15 Description of the timestamp parameters

<table>
<thead>
<tr>
<th>Timestamp parameters</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot</td>
<td>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.</td>
<td>%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.</td>
</tr>
<tr>
<td>date</td>
<td>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.</td>
<td>%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.</td>
</tr>
<tr>
<td>iso</td>
<td>Timestamp format stipulated in ISO 8601. Only logs that are sent to a log host support this parameter.</td>
<td>&lt;189&gt;2003-05-30T06:42:44 Sysname %%10FTPDP/5/FTPD_LOGIN(): User ftp (192.168.1.23) has logged in successfully. 2003-05-30T06:42:44 is a timestamp in the iso format.</td>
</tr>
<tr>
<td>none</td>
<td>No timestamp is included. All logs support this parameter.</td>
<td>% Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully. No timestamp is included.</td>
</tr>
</tbody>
</table>
### 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

<table>
<thead>
<tr>
<th>Task at a glance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform at least one of the following tasks:</td>
</tr>
<tr>
<td>- Outputting logs to the console</td>
</tr>
<tr>
<td>- Outputting logs to the monitor terminal</td>
</tr>
<tr>
<td>- Outputting logs to a log host</td>
</tr>
<tr>
<td>- Outputting logs to the log buffer</td>
</tr>
<tr>
<td>- Saving logs to the log file</td>
</tr>
<tr>
<td>(Optional.) Managing security logs</td>
</tr>
<tr>
<td>(Optional.) Saving diagnostic logs to the diagnostic log file</td>
</tr>
<tr>
<td>(Optional.) Configuring the maximum size of the trace log file</td>
</tr>
<tr>
<td>(Optional.) Setting the minimum storage period for logs</td>
</tr>
<tr>
<td>(Optional.) Enabling synchronous information output</td>
</tr>
<tr>
<td>(Optional.) Enabling duplicate log suppression</td>
</tr>
<tr>
<td>(Optional.) Configuring log suppression for a module</td>
</tr>
<tr>
<td>(Optional.) Disabling an interface from generating link up or link down logs</td>
</tr>
</tbody>
</table>

### Outputting logs to the console

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable the information center.</td>
<td>info-center enable</td>
</tr>
<tr>
<td>3.</td>
<td>Configure an output rule for the console.</td>
<td>info-center source { module-name</td>
</tr>
</tbody>
</table>
### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable the information center.</td>
<td>info-center enable</td>
</tr>
<tr>
<td>3.</td>
<td>Configure an output rule for the monitor terminal.</td>
<td>info-center source { module-name</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Configure the timestamp format.</td>
<td>info-center timestamp { boot</td>
</tr>
<tr>
<td>5.</td>
<td>Return to user view.</td>
<td>quit</td>
</tr>
<tr>
<td>6.</td>
<td>Enable log output to the monitor terminal.</td>
<td>terminal monitor</td>
</tr>
<tr>
<td>7.</td>
<td>Enable the display of debug information on the current terminal.</td>
<td>terminal debugging</td>
</tr>
<tr>
<td>8.</td>
<td>(Optional.) Set the lowest severity level of logs that can be output to the monitor terminal.</td>
<td>terminal logging level severity</td>
</tr>
</tbody>
</table>

### Outputting logs to a log host

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable the information center.</td>
<td>info-center enable</td>
</tr>
</tbody>
</table>
### Outputting logs to the log buffer

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>info-center enable</td>
<td>By default, the information center is enabled.</td>
</tr>
<tr>
<td>3.</td>
<td>info-center logbuffer</td>
<td>By default, log output to the log buffer is enabled.</td>
</tr>
<tr>
<td>4.</td>
<td>info-center logbuffer size buffersize</td>
<td>By default, the log buffer can store 512 logs.</td>
</tr>
<tr>
<td>5.</td>
<td>info-center source { module-name</td>
<td>default } { console</td>
</tr>
<tr>
<td>6.</td>
<td>info-center timestamp { boot</td>
<td>date</td>
</tr>
</tbody>
</table>

### Saving logs to the log file

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable the information center.</td>
<td>info-center enable</td>
</tr>
<tr>
<td>3.</td>
<td>Enable the log file feature.</td>
<td>info-center logfile enable</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Enable log file overwrite-protection.</td>
<td>info-center logfile overwrite-protection [ all-port-powerdown ]</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Configure the maximum size for the log file.</td>
<td>info-center logfile size-quota size</td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Specify the directory to save the log file.</td>
<td>info-center logfile directory dir-name</td>
</tr>
<tr>
<td>7.</td>
<td>Save the logs in the log file buffer to the log file.</td>
<td>• Configure the interval to perform the save operation: info-center logfile frequency freq-sec&lt;br&gt;• Manually save the logs in the log file buffer to the log file: logfile save</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view. system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Enable the information center. info-center enable</td>
<td>By default, the information center is enabled.</td>
</tr>
<tr>
<td>3.</td>
<td>Enable the saving of the security logs to the security log file. info-center security-logfile enable</td>
<td>By default, saving security logs to the security log file is disabled.</td>
</tr>
<tr>
<td>4.</td>
<td>Set the interval at which the system saves security logs. info-center security-logfile frequency freq-sec</td>
<td>By default, the system saves security logs to the security log file every 86400 seconds.</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Set the maximum size of the security log file. info-center security-logfile size-quota size</td>
<td>By default, the maximum size of the security log file is 10 MB.</td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Set the alarm threshold of the security log file usage. info-center security-logfile alarm-threshold usage</td>
<td>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.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display a summary of the security log file.</td>
<td>display security-logfile summary</td>
<td>Available in user view.</td>
</tr>
<tr>
<td>Change the directory of the security log file.</td>
<td>1. system-view 2. info-center security-logfile directory dir-name</td>
<td>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.</td>
</tr>
<tr>
<td>Manually save all the contents in the security log file buffer to the security log file.</td>
<td>security-logfile save</td>
<td>Available in any view.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view, N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Enable the information center.</td>
<td>info-center enable, By default, the information center is enabled.</td>
</tr>
<tr>
<td>3.</td>
<td>Enable saving diagnostic logs to the diagnostic log file.</td>
<td>info-center logfile enable, By default, saving diagnostic logs to the diagnostic log file is enabled.</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Configure the maximum size of the diagnostic log file.</td>
<td>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.</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Specify the directory to save diagnostic log files.</td>
<td>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.</td>
</tr>
<tr>
<td>6.</td>
<td>Save the diagnostic logs in the diagnostic log file buffer to the diagnostic log file.</td>
<td>• Configure the interval to perform the saving operation: info-center diagnostic-logfile frequency freq-sec, The default saving interval is 86400 seconds. The diagnostic-logfile save command is available in any view. • Manually save the diagnostic logs in the buffer to the diagnostic log file: diagnostic-logfile save</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view, N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Set the maximum size of the trace log file.</td>
<td>info-center diagnostic-logfile quota size, By default, the maximum size of the trace log file is 1 MB.</td>
</tr>
</tbody>
</table>
Setting the minimum storage period for logs

Use this feature to set the minimum storage period for logs in the log buffer and log file. This feature ensures that logs will not be overwritten by new logs in a specific period of time.

By default, when the log buffer or log file is full, new logs will automatically overwrite the oldest logs. After the minimum storage period is set, the system identifies the storage period of a log to determine whether to delete the log. The system current time minus a log’s generation time is the log’s storage period.

- If the storage period of a log is shorter than or equal to the minimum storage period, the system does not delete the log. The new log will not be saved.
- If the storage period of a log is longer than the minimum storage period, the system deletes the log to save the new log.

To set the log minimum storage period:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Set the log minimum storage period.</td>
<td>info-center syslog min-age min-age</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable synchronous information output.</td>
<td>info-center synchronous</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enable duplicate log suppression.</td>
<td>info-center logging suppress duplicates</td>
<td>By default, duplicate log suppression is disabled.</td>
</tr>
</tbody>
</table>

Configuring log suppression for a module

This feature suppresses output of logs. You can use this feature to filter out the logs that you are not concerned with.

Perform this task to configure a log suppression rule to suppress output of all logs or logs with a specific mnemonic value for a module.

To configure a log suppression rule for a module:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Configure a log suppression rule for a module.</td>
<td>info-center logging suppress module module-name mnemonic { all</td>
<td>mnemonic-content }</td>
</tr>
</tbody>
</table>

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 specific 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enter interface view.</td>
<td>interface interface-type interface-number</td>
<td>N/A</td>
</tr>
<tr>
<td>3. Disable the interface from generating link up or link down logs.</td>
<td>undo enable log updown</td>
<td>By default, all interfaces generate link up and link down logs when the interface state changes.</td>
</tr>
</tbody>
</table>

Displaying and maintaining information center

Execute **display** commands in any view and **reset** commands in user view.
<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display the diagnostic log file configuration.</td>
<td>display diagnostic-logfile summary</td>
</tr>
<tr>
<td>Display the information of each output destination.</td>
<td>display info-center</td>
</tr>
<tr>
<td>Display the state and the log information of the log buffer.</td>
<td>display logbuffer [ reverse ] [ level severity</td>
</tr>
<tr>
<td>Display a summary of the log buffer.</td>
<td>display logbuffer summary [ level severity</td>
</tr>
<tr>
<td>Display the configuration of the log file.</td>
<td>display logfile summary</td>
</tr>
<tr>
<td>Clear the log buffer.</td>
<td>reset logbuffer</td>
</tr>
</tbody>
</table>

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 19 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 20 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.

```plaintext
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.

![Network diagram](image)

Configuration procedure

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

1. Configure the device:
   ```plaintext
   # 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.
   ```bash
   # mkdir /var/log/Device
   # touch /var/log/Device/info.log
   ``

c. Edit the file syslog.conf in directory /etc/ and add the following contents.
   ```bash
   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.
   ```bash
   # 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 22 Relationship between NMS, agent, and MIB](image)

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 23 is uniquely identified by the OID {1.2.1.1}.

![Figure 23 MIB tree](image)

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.

## 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 acknowledgement but traps do not. Traps are available in SNMPv1, SNMPv2c, and SNMPv3. 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*.

RBAC mode controls access on a per MIB object basis, and VACM mode controls access on a MIB view basis. As a best practice to enhance MIB security, use RBAC mode.
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>(Optional.) Enable the SNMP agent.</td>
<td>snmp-agent</td>
</tr>
<tr>
<td>3.</td>
<td>(Optional.) Configure the system contact.</td>
<td>snmp-agent sys-info contact sys-contact</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Configure the system location.</td>
<td>snmp-agent sys-info location sys-location</td>
</tr>
<tr>
<td>5.</td>
<td>Enable SNMPv1 or SNMPv2c.</td>
<td>snmp-agent sys-info version { all</td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Change the local engine ID.</td>
<td>snmp-agent local-engineid engineid</td>
</tr>
<tr>
<td>7.</td>
<td>(Optional.) Create or update a MIB view.</td>
<td>snmp-agent mib-view { excluded</td>
</tr>
</tbody>
</table>

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.

By default, the system contact is New H3C Technologies Co., Ltd..

By default, the system location is Hangzhou, China.

By default, SNMPv3 is used.

By default, the local engine ID is the company ID plus the device ID.

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.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.</td>
<td>(Method 1.) Create an SNMP community:</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>In VACM mode:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>snmp-agent community { read</td>
<td>write } [ simple</td>
</tr>
<tr>
<td></td>
<td>community-name [ mib-view view-name ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ acl acl-number</td>
<td>acl ipv6 ipv6-acl-number ]</td>
</tr>
<tr>
<td></td>
<td>In RBAC mode:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>snmp-agent community [ simple</td>
<td>cipher ]</td>
</tr>
<tr>
<td></td>
<td>user-role role-name [ acl acl-number</td>
<td>acl ipv6 ipv6-acl-number ]</td>
</tr>
<tr>
<td></td>
<td>(Method 2.) Create an SNMPv1/v2c group, and add users to the group:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a. snmp-agent group { v1</td>
<td>v2c }</td>
</tr>
<tr>
<td></td>
<td>b. snmp-agent usm-user { v1</td>
<td>v2c } user-name group-name</td>
</tr>
<tr>
<td>9.</td>
<td>(Optional.) Create an SNMP context.</td>
<td>By default, no SNMP context is configured on the device.</td>
</tr>
<tr>
<td></td>
<td>snmp-agent context context-name</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>(Optional.) Map an SNMP community to an SNMP context.</td>
<td>By default, no mapping between an SNMP community and an SNMP context exists on the device.</td>
</tr>
<tr>
<td></td>
<td>snmp-agent community-map community-name context context-name</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>(Optional.) Configure the maximum SNMP packet size (in bytes) that the SNMP agent can handle.</td>
<td>By default, the maximum SNMP packet size that the SNMP agent can handle is 1500 bytes.</td>
</tr>
<tr>
<td></td>
<td>snmp-agent packet max-size byte-count</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>Specify the UDP port for receiving SNMP packets.</td>
<td>By default, the device uses UDP port 161 for receiving SNMP packets.</td>
</tr>
<tr>
<td></td>
<td>snmp-agent port port-number</td>
<td></td>
</tr>
</tbody>
</table>

**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

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

To configure SNMPv3 basic parameters:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view N/A</td>
</tr>
<tr>
<td>2.</td>
<td>(Optional.) Enable the SNMP agent.</td>
<td>snmp-agent By default, the SNMP agent is disabled. The SNMP agent is enabled when you use any command that begins with <code>snmp-agent</code> except for the <code>snmp-agent calculate-password</code> command.</td>
</tr>
<tr>
<td>3.</td>
<td>(Optional.) Configure the system contact.</td>
<td>snmp-agent sys-info contact sys-contact By default, the system contact is New H3C Technologies Co., Ltd.</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Configure the system location.</td>
<td>snmp-agent sys-info location sys-location By default, the system location is Hangzhou, China.</td>
</tr>
</tbody>
</table>
| 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. |
| 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. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td>(Optional.) Configure a remote engine ID.</td>
<td><strong>snmp-agent remote</strong> *(ip-address</td>
</tr>
<tr>
<td>8.</td>
<td>(Optional.) Create or update a MIB view.</td>
<td><strong>snmp-agent mib-view</strong> *(excluded</td>
</tr>
<tr>
<td>9.</td>
<td>Create an SNMPv3 group.</td>
<td>• In non-FIPS mode: <strong>snmp-agent group v3</strong> <strong>group-name</strong> *(authentication</td>
</tr>
<tr>
<td>10.</td>
<td>(Optional.) Calculate a digest for the ciphertext key converted from a plaintext key.</td>
<td>• In non-FIPS mode: <strong>snmp-agent calculate-password</strong> <strong>plain-password</strong> *(mode 3desmd5</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 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  
| 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, no SNMP context is configured on the device.  
By default, the maximum SNMP packet size that the SNMP agent can handle is 1500 bytes. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.</td>
<td><code>snmp-agent port port-number</code></td>
<td>By default, the device uses UDP port 161 for receiving SNMP packets.</td>
</tr>
</tbody>
</table>

### 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.

To configure SNMP logging:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>`snmp-agent log { all</td>
<td>authfail</td>
</tr>
<tr>
<td>3.</td>
<td><code>snmp-agent trap log</code></td>
<td>By default, SNMP notification logging is disabled.</td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
</tbody>
</table>
### 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 acknowledgement 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enable</td>
<td><code>snmp-agent trap enable</code></td>
<td>By default, SNMP configuration notifications,</td>
</tr>
<tr>
<td></td>
<td><code>configuration</code></td>
<td>standard notifications, and system notifications, and system notifications are enabled. Whether other SNMP notifications are enabled varies by modules.</td>
</tr>
<tr>
<td></td>
<td><code>protocol</code></td>
<td>system</td>
</tr>
<tr>
<td></td>
<td><code>standard</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>authentication</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>coldstart</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>linkdown</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>linkup</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>warmstart</code></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td><code>interface interface-type interface-number</code></td>
<td>N/A</td>
</tr>
<tr>
<td>4. Enable</td>
<td><code>enable snmp trap updown</code></td>
<td>By default, link state notifications are enabled.</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 2.   | **Configure a target host.**<br>• Send traps to the target host:<br>In non-FIPS mode:<br>`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 ] ]`<br>In FIPS mode:<br>`snmp-agent target-host trap address udp-domain { target-host | ipv6 target-host } [ udp-port port-number ] params securityname security-string v3 { authentication | privacy }`<br>• Send informs to the target host:<br>In non-FIPS mode:<br>`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 ] ]`<br>In FIPS mode:<br>`snmp-agent target-host inform address udp-domain { target-host | ipv6 target-host } [ udp-port port-number ] params securityname security-string v3 { authentication | privacy }`<br>By default, no target host is configured. |<br>| 3.   | (Optional.) Configure a source address for notifications.<br>`snmp-agent { inform | trap } source interface-type interface-number`<br>By default, SNMP uses the IP address of the outgoing routed interface as the source IP address. |<br>| 4.   | (Optional.) Enable extended linkUp/linkDown notifications.<br>`snmp-agent trap if-mib link extended`<br>By default, the SNMP agent sends standard linkUp/linkDown notifications. |<br>| 5.   | (Optional.) Configure the notification queue size.<br>`snmp-agent trap queue-size size`<br>By default, the notification queue can hold 100 notification messages. |<br>| 6.   | (Optional.) Configure the notification lifetime.<br>`snmp-agent trap life seconds`<br>The default notification lifetime is 120 seconds. |<br>### Displaying the SNMP settings

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display SNMP agent system information, including the contact, physical location, and SNMP version.</td>
<td>`display snmp-agent sys-info [ contact</td>
</tr>
<tr>
<td>Display SNMP agent statistics.</td>
<td><code>display snmp-agent statistics</code></td>
</tr>
<tr>
<td>Display the local engine ID.</td>
<td><code>display snmp-agent local-engineid</code></td>
</tr>
<tr>
<td>Task</td>
<td>Command</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Display SNMP group information.</td>
<td>display snmp-agent group [ group-name ]</td>
</tr>
<tr>
<td>Display remote engine IDs.</td>
<td>display snmp-agent remote [ ip-address</td>
</tr>
<tr>
<td>Display basic information about the notification queue.</td>
<td>display snmp-agent trap queue</td>
</tr>
<tr>
<td>Display the modules that can generate notifications and their notification status (enable or disable).</td>
<td>display snmp-agent trap-list</td>
</tr>
<tr>
<td>Display SNMPv3 user information.</td>
<td>display snmp-agent usm-user [ engineid engineid</td>
</tr>
<tr>
<td>Display SNMPv1 or SNMPv2c community information. (This command is not supported in FIPS mode.)</td>
<td>display snmp-agent community [ read</td>
</tr>
<tr>
<td>Display MIB view information.</td>
<td>display snmp-agent mib-view [ exclude</td>
</tr>
<tr>
<td>Display SNMP MIB node information.</td>
<td>display snmp-agent mib-node [ details</td>
</tr>
<tr>
<td>Display an SNMP context.</td>
<td>display snmp-agent context [ context-name ]</td>
</tr>
</tbody>
</table>

**SNMP configuration examples**

**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 24, 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 24 Network diagram**

![Network Diagram](image)

**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 25, 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&!`.  

76
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:
SNMPv3 in RBAC mode configuration example

Network requirements

As shown in Figure 26, 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 26 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 hh3cUIMgt 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:
   o Specify SNMPv3.
   o Create the SNMPv3 user managev3user.
   o Enable both authentication and privacy functions.
   o Use SHA-1 for authentication and AES for encryption.
   o Set the authentication key to 123456TESTauth&! and the privacy key to 123456TESTencr&!.
   o 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=h3c 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=sysName-
 e(1.3.6.1.2.1.1.5.0)-Value=h3c; Permission denied.

# Log in to the agent. You can see a notification on the NMS.
hh3cLogIn 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) hh3cLogIn
    Binding #3: hh3cTerminalUserName.0 *** (octets) testuser
                  [74.65.73.74.75.73.65.72 hex]
    Binding #4: hh3cTerminalSource.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".

H3C 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, H3C implements the statistics group, history group, event group, alarm group, probe configuration group, and user history group. H3C 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 27.

**Figure 27 Rising and falling alarm events**

![Diagram showing rising and falling alarm events](image)

**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 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view. <strong>system-view</strong></td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Enter Ethernet interface view. <strong>interface interface-type interface-number</strong></td>
<td>N/A</td>
</tr>
<tr>
<td>3.</td>
<td>Create an entry for the interface in the RMON statistics table. <strong>rmon statistics entry-number [ owner text ]</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter Ethernet interface view.</td>
<td>interface interface-type interface-number</td>
</tr>
<tr>
<td>3.</td>
<td>Create an entry for the interface in the RMON history control table.</td>
<td>rmon history entry-number buckets number interval sampling-interval [ owner text ]</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Entry</th>
<th>Parameters to be compared</th>
<th>Maximum number of entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>• Event description (description string)</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>• Event type (log, trap, logtrap, or none)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Community name (security-string)</td>
<td></td>
</tr>
<tr>
<td>Alarm</td>
<td>• Alarm variable (alarm-variable)</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>• Sampling interval (sampling-interval)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Sample type (absolute or delta)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Rising threshold (threshold-value1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Falling threshold (threshold-value2)</td>
<td></td>
</tr>
<tr>
<td>Private alarm</td>
<td>• Alarm variable formula (prialarm-formula)</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>• Sampling interval (sampling-interval)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Sample type (absolute or delta)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Rising threshold (threshold-value1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Falling threshold (threshold-value2)</td>
<td></td>
</tr>
</tbody>
</table>

To configure the RMON alarm function:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>(Optional.) Create an event entry in the event table.</td>
<td>rmon event entry-number [ description string ] [ log</td>
</tr>
</tbody>
</table>
3. Create an entry in the alarm table or private alarm table.

- Create an entry in the alarm table: 
  \[
  \text{rmon alarm entry-number alarm-variable sampling-interval} \{ \text{absolute | delta} \} \\
  \{ \text{startup-alarm} \{ \text{falling | rising | rising-falling} \} \} \text{rising-threshold} \text{threshold-value1 event-entry1} \\
  \text{falling-threshold} \text{threshold-value2 event-entry2} \{ \text{owner text} \}
  \]

- Create an entry in the private alarm table: 
  \[
  \text{rmon prialarm entry-number prialarm-formula prialarm-des sampling-interval} \{ \text{absolute | delta} \} \{ \text{startup-alarm} \{ \text{falling | rising | rising-falling} \} \} \text{rising-threshold} \text{threshold-value1 event-entry1} \\
  \text{falling-threshold} \text{threshold-value2 event-entry2} \text{entrytype} \{ \text{forever | cycle cycle-period} \} \{ \text{owner text} \}
  \]

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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display RMON statistics.</td>
<td><code>display rmon statistics [interface-type interface-number]</code></td>
</tr>
<tr>
<td>Display RMON history control entries and history samples.</td>
<td><code>display rmon history [interface-type interface-number]</code></td>
</tr>
<tr>
<td>Display RMON alarm entries.</td>
<td><code>display rmon alarm [entry-number]</code></td>
</tr>
<tr>
<td>Display RMON private alarm entries.</td>
<td><code>display rmon prialarm [entry-number]</code></td>
</tr>
<tr>
<td>Display RMON event entries.</td>
<td><code>display rmon event [entry-number]</code></td>
</tr>
<tr>
<td>Display log information for event entries.</td>
<td><code>display rmon eventlog [entry-number]</code></td>
</tr>
</tbody>
</table>

RMON configuration examples

Ethernet statistics group configuration example

Network requirements

As shown in Figure 28, create an RMON Ethernet statistics entry on the device to gather cumulative traffic statistics for GigabitEthernet 1/0/1.

**Figure 28 Network diagram**

![Network Diagram](image)
Configuration procedure

# Create an RMON Ethernet statistics entry for GigabitEthernet 1/0/1.
<Sysname> system-view
[Sysname] interface gigabitethernet 1/0/1
[Sysname-GigabitEthernet1/0/1] rmon statistics 1 owner user1

# Display statistics collected by the RMON agent for GigabitEthernet 1/0/1.
<Sysname> display rmon statistics gigabitethernet 1/0/1
EtherStatsEntry 1 owned by user1 is VALID.
  Interface : 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:
   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 29, create an RMON history control entry on the device to sample traffic statistics for GigabitEthernet 1/0/1 every minute.

Figure 29 Network diagram

Configuration procedure

# Create an RMON history control entry to sample traffic statistics every minute for GigabitEthernet 1/0/1. Retain up to eight samples for the interface in the history statistics table.
<Sysname> system-view
[Sysname] interface gigabitethernet 1/0/1
[Sysname-GigabitEthernet1/0/1] rmon history 1 buckets 8 interval 60 owner user1

# Display the history statistics collected for GigabitEthernet 1/0/1.
[Sysname-GigabitEthernet1/0/1] display rmon history
HistoryControlEntry 1 owned by user1 is VALID
   Sampled interface     : 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 30, configure the device to monitor the incoming traffic statistic on 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 30 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.

```bash
<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 GigabitEthernet 1/0/1.
[Sysname-GigabitEthernet1/0/1] rmon statistics 1 owner user1

# 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-GigabitEthernet1/0/1] rmon event 1 trap public owner user1
[Sysname-GigabitEthernet1/0/1] 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 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 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 GigabitEthernet 1/0/1.
<Sysname> display rmon statistics gigabitethernet 1/0/1
EtherStatsEntry 1 owned by user1 is VALID.
  Interface : 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 31, 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 31 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.
• A UDP tracert operation determines the routing path from the source to the destination. The number of the probes to each hop is set by using the probe count 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 32 Collaboration

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 display results or send trap messages to the NMS when the performance metrics that an NQA operation gathers violate the specified thresholds.

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

<table>
<thead>
<tr>
<th>Performance metric</th>
<th>NQA operation types that can gather the metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probe duration</td>
<td>All NQA operation types except UDP jitter, path jitter, and voice</td>
</tr>
<tr>
<td>Number of probe failures</td>
<td>All NQA operation types except UDP jitter, path jitter, and voice</td>
</tr>
<tr>
<td>Round-trip time</td>
<td>UDP jitter and voice</td>
</tr>
<tr>
<td>Number of discarded packets</td>
<td>UDP jitter and voice</td>
</tr>
<tr>
<td>One-way jitter (source-to-destination and destination-to-source)</td>
<td>UDP jitter and voice</td>
</tr>
<tr>
<td>One-way latency (source-to-destination and destination-to-source)</td>
<td>UDP jitter and voice</td>
</tr>
<tr>
<td>Calculated Planning Impairment Factor (ICPIF) (see &quot;Configuring the voice operation&quot;)</td>
<td>Voice</td>
</tr>
<tr>
<td>Mean Opinion Scores (MOS) (see &quot;Configuring the voice operation&quot;)</td>
<td>Voice</td>
</tr>
</tbody>
</table>

NQA configuration task list

<table>
<thead>
<tr>
<th>Tasks at a glance</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring the NQA server</td>
<td>Required for TCP, UDP echo, UDP jitter, and voice operations.</td>
</tr>
</tbody>
</table>

(Required.) Use one of the following methods:
- Configure NQA operations:
  - Enabling the NQA client
  - Configuring NQA operations on the NQA client
  - Scheduling the NQA operation 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa server enable</td>
<td>By default, the NQA server is disabled.</td>
</tr>
</tbody>
</table>
3. Configure a TCP or UDP listening service.

   - **TCP listening service:**
     \[\text{nqa server tcp-connect ip-address port-number [tos tos]}\]
   
   - **UDP listening service:**
     \[\text{nqa server udp-echo ip-address port-number [tos tos]}\]

   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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable the NQA client.</td>
<td>nqa agent enable</td>
</tr>
</tbody>
</table>

### Configuring NQA operations on the NQA client

#### NQA operation configuration task list

**Tasks at a glance**

- **(Required.)** Configure at least one of the NQA operations:
  - 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 SNMIP operation
  - Configuring the TCP operation
  - Configuring the UDP echo operation
  - Configuring the UDP tracer 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

### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td><code>nqa entry admin-name operation-tag</code></td>
</tr>
<tr>
<td>3.</td>
<td>Specify the ICMP echo type and enter its view.</td>
<td><code>type icmp-echo</code></td>
</tr>
<tr>
<td>4.</td>
<td>Specify the destination address of ICMP echo requests.</td>
<td><code>destination ip ip-address</code></td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Specify the payload size in each ICMP echo request.</td>
<td><code>data-size size</code></td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Specify the string to be filled in the payload of each ICMP echo request.</td>
<td><code>data-fill string</code></td>
</tr>
<tr>
<td>7.</td>
<td>(Optional.) Specify the output interface for ICMP echo requests.</td>
<td><code>out interface interface-type interface-number</code></td>
</tr>
<tr>
<td>8.</td>
<td>(Optional.) Specify the source IP address of ICMP echo requests.</td>
<td>- Specify the IP address of the specified interface as the source IP address: <code>source interface interface-type interface-number</code> - Specify the source IP address: <code>source ip ip-address</code></td>
</tr>
<tr>
<td>9.</td>
<td>(Optional.) Specify the next hop for ICMP echo requests.</td>
<td><code>next-hop ip-address</code></td>
</tr>
</tbody>
</table>

**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 for 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td>nqa entry admin-name operation-tag</td>
</tr>
<tr>
<td>3.</td>
<td>Specify the DHCP type and enter its view.</td>
<td>type dhcp</td>
</tr>
<tr>
<td>4.</td>
<td>Specify the IP address of the DHCP server as the destination IP address of DHCP packets.</td>
<td>destination ip ip-address</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Specify the output interface for ICMP echo requests.</td>
<td>out interface interface-type interface-number</td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Specify the source IP address of DHCP request packets.</td>
<td>source ip ip-address</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td>nqa entry admin-name operation-tag</td>
</tr>
</tbody>
</table>
### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa entry admin-name operation-tag</td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td>3.</td>
<td>type ftp</td>
<td>N/A</td>
</tr>
</tbody>
</table>
| 4.   | 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.   | 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.   | operation { get | put } | By default, the FTP operation type is `get`, which means obtaining files from the FTP server. |
| 7.   | username | By default, no FTP login username is configured. |
### Configuring the FTP Operation

An FTP operation measures the time for the NQA client to obtain data from an HTTP server.

To configure an FTP operation:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.</td>
<td>Specify an FTP login password.</td>
<td>password { cipher</td>
</tr>
<tr>
<td></td>
<td>By default, no FTP login password is configured.</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>(Optional.) Specify the name of a file to be transferred.</td>
<td>filename file-name</td>
</tr>
<tr>
<td></td>
<td>By default, no file is specified. This step is required if you perform the put operation.</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>Set the data transmission mode.</td>
<td>mode { active</td>
</tr>
<tr>
<td></td>
<td>The default mode is active.</td>
<td></td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td>nqa entry admin-name operation-tag</td>
</tr>
<tr>
<td></td>
<td>By default, no NQA operation is created.</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Specify the HTTP type and enter its view.</td>
<td>type http</td>
</tr>
<tr>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Specify the URL of the destination HTTP server.</td>
<td>url url</td>
</tr>
<tr>
<td></td>
<td>By default, no URL is specified for the destination HTTP server. Enter the URL in one of the following formats:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://host/resource">http://host/resource</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://host:port/resource">http://host:port/resource</a></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Specify an HTTP login username.</td>
<td>username username</td>
</tr>
<tr>
<td></td>
<td>By default, no HTTP login username is specified.</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Specify an HTTP login password.</td>
<td>password { cipher</td>
</tr>
<tr>
<td></td>
<td>By default, no HTTP login password is specified.</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>(Optional.) Specify the source IP address of request packets.</td>
<td>source ip ip-address</td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>(Optional.) Specify the HTTP version.</td>
<td>version { v1.0</td>
</tr>
<tr>
<td></td>
<td>By default, HTTP 1.0 is used.</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>(Optional.) Specify the HTTP operation type.</td>
<td>operation { get</td>
</tr>
<tr>
<td></td>
<td>By default, the HTTP operation type is get, which means obtaining data from the HTTP server.</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>(Optional.) Enter raw request view.</td>
<td>raw-request</td>
</tr>
<tr>
<td></td>
<td>Every time you enter raw request view, the previously configured content of the HTTP request is removed.</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>(Optional.) Specify the content of a GET request for the HTTP operation.</td>
<td>Enter or paste the content.</td>
</tr>
<tr>
<td></td>
<td>By default, no contents are specified. This step is required for the raw operation.</td>
<td></td>
</tr>
</tbody>
</table>
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.
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa entry admin-name operation-tag</td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td>3.</td>
<td>type udp-jitter</td>
<td>N/A</td>
</tr>
<tr>
<td>4.</td>
<td>destination ip ip-address</td>
<td>By default, no destination IP address is specified. The destination IP address must be the same as that of the listening service on the NQA server.</td>
</tr>
<tr>
<td>5.</td>
<td>destination port port-number</td>
<td>By default, no destination port number is specified. The destination port must be the same as that of the listening service on the NQA server.</td>
</tr>
<tr>
<td>6.</td>
<td>source port port-number</td>
<td>By default, no source port number is specified.</td>
</tr>
<tr>
<td>7.</td>
<td>data-size size</td>
<td>The default setting is 100 bytes.</td>
</tr>
<tr>
<td>8.</td>
<td>data-fill string</td>
<td>The default setting is the hexadecimal number 00010203040506070809.</td>
</tr>
</tbody>
</table>
### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa entry admin-name operation-tag</td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td>3.</td>
<td>type snmp</td>
<td>N/A</td>
</tr>
<tr>
<td>4.</td>
<td>destination ip ip-address</td>
<td>By default, no destination IP address is specified.</td>
</tr>
<tr>
<td>5.</td>
<td>source port port-number</td>
<td>By default, no source port number is specified.</td>
</tr>
<tr>
<td>6.</td>
<td>source ip ip-address</td>
<td>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.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa entry admin-name operation-tag</td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td>3.</td>
<td>type tcp</td>
<td>N/A</td>
</tr>
<tr>
<td>4.</td>
<td>destination ip ip-address</td>
<td>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.</td>
</tr>
<tr>
<td>5.</td>
<td>destination port port-number</td>
<td>By default, no destination port number is configured. The destination port number must be the same as that of the listening service on the NQA server.</td>
</tr>
<tr>
<td>6.</td>
<td>source ip ip-address</td>
<td>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.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa entry admin-name operation-tag</td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td>3.</td>
<td>type udp-echo</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Configuring the UDP tracert operation

The UDP tracert operation is not supported in IPv6 networks. To determine the routing path that the IPv6 packets traverse from the source to the destination, use the `tracert ipv6` command. For more information about the command, see `Network Management and Monitoring Command Reference`.

The UDP tracert operation determines the routing path from the source device to the destination device.

Before you configure the UDP tracert operation, perform the following tasks:

- Enable sending ICMP time exceeded messages on the intermediate devices between the source and destination devices. If the intermediate devices are H3C devices, use the `ip ttl-expires enable` command.
- Enable sending ICMP destination unreachable messages on the destination device. If the destination device is an H3C device, use the `ip unreachables enable` command.

For more information about the `ip ttl-expires enable` and `ip unreachables enable` commands, see `Layer 3—IP Services Command Reference`.

To configure the UDP tracert operation:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa entry admin-name operation-tag</td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>3.</td>
<td>Specify the UDP tracert operation type and enter its view.</td>
<td>type udp-tracert</td>
</tr>
<tr>
<td>4.</td>
<td>Specify the destination address of UDP packets.</td>
<td>destination ip ip-address</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Specify the destination port of UDP packets.</td>
<td>destination port port-number</td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Set the payload size for each UDP packet.</td>
<td>data-size size</td>
</tr>
<tr>
<td>7.</td>
<td>(Optional.) Enable the no-fragmentation feature.</td>
<td>no-fragment enable</td>
</tr>
<tr>
<td>8.</td>
<td>(Optional.) Set the maximum number of consecutive probe failures.</td>
<td>max-failure value</td>
</tr>
<tr>
<td>9.</td>
<td>(Optional.) Set the TTL value for UDP packets in the start round of the UDP tracert operation.</td>
<td>init-ttl value</td>
</tr>
<tr>
<td>10.</td>
<td>(Optional.) Specify an output interface for UDP packets.</td>
<td>out interface interface-type interface-number</td>
</tr>
<tr>
<td>11.</td>
<td>(Optional.) Specify the source port of UDP packets.</td>
<td>source port port-number</td>
</tr>
</tbody>
</table>
| 12.  | (Optional.) Specify the source IP address of UDP packets. | • 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 packets 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 local interface must be up. Otherwise, no probe 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td><code>nqa entry admin-name</code></td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td></td>
<td><code>operation-tag</code></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td><code>type voice</code></td>
<td>N/A</td>
</tr>
<tr>
<td>4.</td>
<td><code>destination ip ip-address</code></td>
<td>By default, no destination IP address is configured. The destination IP address must be the same as that of the listening service on the NQA server.</td>
</tr>
<tr>
<td>5.</td>
<td><code>destination port port-number</code></td>
<td>By default, no destination port number is configured. The destination port must be the same as that of the listening service on the NQA server.</td>
</tr>
<tr>
<td>6.</td>
<td>`codec-type { g711a</td>
<td>g711u</td>
</tr>
</tbody>
</table>
### Configuring the DLSw operation

The DLSw operation measures the response time of a DLSw device.

To configure the DLSw operation:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td>nqa entry admin-name operation-tag</td>
</tr>
<tr>
<td>3.</td>
<td>Specify the DLSw type and enter its view.</td>
<td>type dlsw</td>
</tr>
</tbody>
</table>
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 H3C devices, use the `ip ttl-expires enable` command.
- Enable sending ICMP destination unreachable messages on the destination device. If the destination device is an H3C device, use the `ip unreachable enable` command.

For more information about the `ip ttl-expires enable` and the `ip unreachable enable` command, see Layer 3—IP Services Command Reference.

To configure the path jitter operation:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td>3.</td>
<td>Specify the path jitter type and enter its view.</td>
<td>N/A</td>
</tr>
<tr>
<td>4.</td>
<td>Specify the destination address of ICMP echo requests.</td>
<td>By default, no destination IP address is specified.</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Specify the payload size in each ICMP echo request.</td>
<td>The default setting is 100 bytes.</td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Specify the string to be filled in the payload of each ICMP echo request.</td>
<td>The default setting is the hexadecimal number 00010203040506070809.</td>
</tr>
<tr>
<td>7.</td>
<td>(Optional.) Specify the source IP address of ICMP echo requests.</td>
<td>By default, no source IP address is specified.</td>
</tr>
<tr>
<td>8.</td>
<td>(Optional.) Specify the number of ICMP echo requests to be sent in a path jitter operation.</td>
<td>The default setting is 10.</td>
</tr>
<tr>
<td>9.</td>
<td>(Optional.) Specify the interval for sending ICMP echo requests.</td>
<td>The default setting is 20 milliseconds.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa entry admin-name operation-tag</td>
<td>By default, no NQA operation is created.</td>
</tr>
<tr>
<td>3.</td>
<td>type { dhcp</td>
<td>disw</td>
</tr>
<tr>
<td>4.</td>
<td>description text</td>
<td>By default, no description is configured.</td>
</tr>
<tr>
<td>5.</td>
<td>frequency interval</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 6.   | probe count times | By default:  
  • In an UDP tracert operation, the NQA client performs three probes to each hop to the destination.  
  • In other types of operations, the NQA client performs one probe to the destination per operation.  
  This command is not available for the path jitter and voice operations. Each of these operations performs only one probe. |
### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td>nqa entry admin-name operation-tag</td>
</tr>
<tr>
<td>3.</td>
<td>Specify an NQA operation type and enter its view.</td>
<td>type { dhcp</td>
</tr>
<tr>
<td>4.</td>
<td>Configure a reaction entry.</td>
<td>reaction item-number checked-element probe-fail threshold-type consecutive consecutivenumber occurrences action-type trigger-only</td>
</tr>
<tr>
<td>5.</td>
<td>Exit to system view.</td>
<td>quit</td>
</tr>
<tr>
<td>6.</td>
<td>Associate Track with NQA.</td>
<td>See High Availability Configuration Guide.</td>
</tr>
<tr>
<td>7.</td>
<td>Associate Track with an application module.</td>
<td>See High Availability Configuration Guide.</td>
</tr>
</tbody>
</table>
Configuring threshold monitoring

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, a monitored element, a threshold type, and an action to be triggered are configured 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td><code>system-view</code></td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td><code>nqa entry admin-name operation-tag</code></td>
</tr>
<tr>
<td>3.</td>
<td>Enter NQA operation view.</td>
<td>`type { dhcp</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 4.   | **reaction trap** { **path-change** | **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 only support the **test-complete** keyword. The following parameters are not available for the UDP tracert operation:
|      |         | • The **probe-failure**
|      |         | consecutive-probe-failures option.
|      |         | • The **cumulate-probe-failures** argument. |

Enable sending traps to the NMS when specific conditions are met.

By default, no traps are sent to the NMS.

The UDP jitter and voice operations only support the **test-complete** keyword.

The following parameters are not available for the UDP tracert operation:

- The **probe-failure** consecutive-probe-failures option.
- The **cumulate-probe-failures** argument.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
</table>
| 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 supported in 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.

A statistics group is deleted when its hold time expires. When the maximum number of statistics groups is reached, to save a new statistics group, the oldest statistics group is deleted.

If you use the `frequency` command to set the interval between two consecutive operations to 0, only one operation is performed. No statistics group information is generated.

To configure the NQA statistics collection function:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td><code>nqa entry admin-name operation-tag</code></td>
</tr>
<tr>
<td>3.</td>
<td>Specify an NQA operation type and enter its view.</td>
<td>`type { dhcp</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Specify the interval for collecting the statistics.</td>
<td><code>statistics interval interval</code></td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Specify the maximum number of statistics groups that can be saved.</td>
<td><code>statistics max-group number</code></td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Specify the hold time of statistics groups.</td>
<td><code>statistics hold-time hold-time</code></td>
</tr>
</tbody>
</table>

Configuring the saving of NQA history records

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA operation and enter NQA operation view.</td>
<td><code>nqa entry admin-name operation-tag</code></td>
</tr>
<tr>
<td>3.</td>
<td>Enter NQA operation type view.</td>
<td>`type { dhcp</td>
</tr>
<tr>
<td>4.</td>
<td>Enable the saving of history records for the NQA operation.</td>
<td>history-record enable</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Set the lifetime of history records.</td>
<td>history-record keep-time keep-time</td>
</tr>
</tbody>
</table>
### 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 only affects the NQA operations that have not started.

To schedule the NQA operation on the NQA client:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>2. Specify the scheduling parameters for an NQA operation.</td>
<td>`nqa schedule admin-name operation-tag start-time { hh:mm:ss</td>
<td>yyyy/mm/dd</td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>2. Create an ICMP template and enter its view.</td>
<td><code>nqa template icmp name</code></td>
<td>N/A</td>
</tr>
</tbody>
</table>
### 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Create a DNS template and enter DNS template view.</td>
<td>nqa template dns name N/A</td>
</tr>
<tr>
<td>3.</td>
<td>(Optional.) Specify the destination IPv4 or IPv6 address of DNS packets.</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>4. (Optional.) Configure the destination port number for the operation.</td>
<td>destination port port-number</td>
<td>By default, the destination port number is 53.</td>
</tr>
<tr>
<td>5. Specify the domain name that needs to be translated.</td>
<td>resolve-target domain-name</td>
<td>By default, no domain name is specified.</td>
</tr>
<tr>
<td>6. Configure the domain name resolution type.</td>
<td>resolve-type { A</td>
<td>AAAA }</td>
</tr>
<tr>
<td>7. (Optional.) Specify the source IPv4 or IPv6 address for the probe packets.</td>
<td>• IPv4 address: source ip ip-address • IPv6 address: source ipv6 ipv6-address</td>
<td>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.</td>
</tr>
<tr>
<td>8. (Optional.) Configure the source port for probe packets.</td>
<td>source port port-number</td>
<td>By default, no source port number is configured.</td>
</tr>
<tr>
<td>9. (Optional.) Specify the IPv4 or IPv6 address that is expected to be returned.</td>
<td>• IPv4 address: expect ip ip-address • IPv6 address: expect ipv6 ipv6-address</td>
<td>By default, no expected IP address is specified.</td>
</tr>
</tbody>
</table>

### Configuring the TCP template

A feature that uses the TCP template performs the TCP operation to determine 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 only tests 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Create a TCP template and enter its view.</td>
<td>nqa template tcp name</td>
<td>N/A</td>
</tr>
<tr>
<td>3. (Optional.) Specify the destination IPv4 or IPv6 address of the operation.</td>
<td>• IPv4 address: destination ip ip-address • IPv6 address: destination ipv6 ipv6-address</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Configuring the UDP template

A feature that uses the UDP template performs the UDP operation to test the following items:

- Reachability of a specific port on the NQA server.
- Availability of the requested service on the NQA server.

In UDP template view, you can specify the expected data to be returned. If you do not specify the expected data, the UDP operation tests only whether the client can receive the response packet from the server.

The UDP operation requires both the NQA server and the NQA client. Before you perform a UDP 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 template:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create a UDP template and enter its view.</td>
<td>nqa template udp name</td>
</tr>
<tr>
<td>3.</td>
<td>(Optional.) Specify the destination IP address of the operation.</td>
<td>destination IP address of the operation.</td>
</tr>
</tbody>
</table>

- IPv4 address:
  - destination ip ip-address
- IPv6 address:
  - destination ipv6 ipv6-address

By default, no expected data is configured. The expected data is checked only when you configure both the data-fill command and the expect-data command.
### Configuring the HTTP Template

A feature that uses the HTTP template performs the HTTP operation to measure the time it takes for 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>nqa template http name</td>
<td>N/A</td>
</tr>
<tr>
<td>3.</td>
<td>url url</td>
<td>By default, no URL is specified for the destination HTTP server. Enter the URL in one of the following formats: • <a href="http://host/resource">http://host/resource</a> • <a href="http://host:port/resource">http://host:port/resource</a></td>
</tr>
<tr>
<td>4.</td>
<td>username username</td>
<td>By default, no HTTP login username is specified.</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>5. Specify an HTTP login password.</td>
<td>`password { cipher</td>
<td>simple } password`</td>
</tr>
<tr>
<td>6. (Optional.) Specify the HTTP version.</td>
<td>`version { v1.0</td>
<td>v1.1 }`</td>
</tr>
<tr>
<td>7. (Optional.) Specify the HTTP operation type.</td>
<td>`operation { get</td>
<td>post</td>
</tr>
<tr>
<td>8. (Optional.) Enter raw request view.</td>
<td><code>raw-request</code></td>
<td>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.</td>
</tr>
<tr>
<td>9. (Optional.) Enter or paste the content of the GET request for the HTTP operation.</td>
<td>N/A</td>
<td>This step is required for the raw operation. By default, no contents are specified.</td>
</tr>
<tr>
<td>10. (Optional.) Save the input and exit to HTTP template view.</td>
<td><code>quit</code></td>
<td>N/A</td>
</tr>
</tbody>
</table>
| 11. (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. |
| 12. (Optional.) Configure the expected status codes. | `expect status status-list` | By default, no expected status code is configured. |
| 13. (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 for 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>2. Create an FTP template and enter its view.</td>
<td><code>nqa template ftp name</code></td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Configuring optional parameters for the NQA template

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create an NQA template and enter its view.</td>
<td>nqa template { dns</td>
</tr>
<tr>
<td>3.</td>
<td>Configure a description.</td>
<td>description text</td>
</tr>
<tr>
<td>4.</td>
<td>Specify the interval at which the NQA operation repeats.</td>
<td>frequency interval</td>
</tr>
<tr>
<td>5.</td>
<td>Specify the probe timeout time.</td>
<td>probe timeout timeout</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>6.</td>
<td>Specify the TTL for probe packets.</td>
<td>ttl value</td>
</tr>
<tr>
<td>7.</td>
<td>(Optional.) Specify the ToS value in the IP header for probe packets.</td>
<td>tos value</td>
</tr>
<tr>
<td>8.</td>
<td>Configure the number of consecutive successful probes that lead to a successful operation.</td>
<td>reaction trigger probe-pass count</td>
</tr>
<tr>
<td>9.</td>
<td>Configure the number of consecutive probe failures that lead to an operation failure.</td>
<td>reaction trigger probe-fail count</td>
</tr>
</tbody>
</table>

### Displaying and maintaining NQA

Execute **display** commands in any view.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display history records of NQA operations.</td>
<td>display nqa history [ admin-name operation-tag ]</td>
</tr>
<tr>
<td>Display the current monitoring results of reaction entries.</td>
<td>display nqa reaction counters [ admin-name operation-tag item-number ]</td>
</tr>
<tr>
<td>Display the most recent result of the NQA operation.</td>
<td>display nqa result [ admin-name operation-tag ]</td>
</tr>
<tr>
<td>Display NQA statistics.</td>
<td>display nqa statistics [ admin-name operation-tag ]</td>
</tr>
<tr>
<td>Display NQA server status.</td>
<td>display nqa server status</td>
</tr>
</tbody>
</table>
NQA configuration examples

ICMP echo operation configuration example

Network requirements

As shown in Figure 33, 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 33 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 and configure the maximum number of history records that can be saved as 10.
[DeviceA-nqa-admin-test1-icmp-echo] history-record enable
[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:

<table>
<thead>
<tr>
<th>Index</th>
<th>Response</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>370</td>
<td>3</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.2</td>
</tr>
<tr>
<td>369</td>
<td>3</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.2</td>
</tr>
<tr>
<td>368</td>
<td>3</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.2</td>
</tr>
<tr>
<td>367</td>
<td>5</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.2</td>
</tr>
<tr>
<td>366</td>
<td>3</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.2</td>
</tr>
<tr>
<td>365</td>
<td>3</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.2</td>
</tr>
<tr>
<td>364</td>
<td>3</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.1</td>
</tr>
<tr>
<td>363</td>
<td>2</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.1</td>
</tr>
<tr>
<td>362</td>
<td>3</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.1</td>
</tr>
<tr>
<td>361</td>
<td>2</td>
<td>Succeeded</td>
<td>2007-08-23 15:00:01.1</td>
</tr>
</tbody>
</table>

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 34, configure a DHCP operation to test the time required for Switch A to obtain an IP address from the DHCP server.
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 Switch A uses 512 milliseconds to obtain an IP address from the DHCP server.

DNS operation configuration example

Network requirements

As shown in Figure 35, configure a DNS operation to test whether Device A can perform address resolution through the DNS server and test the resolution time.
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:
<table>
<thead>
<tr>
<th>Index</th>
<th>Response</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62</td>
<td>Succeeded</td>
<td>2011-11-10 10:49:37.3</td>
</tr>
</tbody>
</table>
```

The output shows that Device A uses 62 milliseconds to translate domain name **host.com** into an IP address.
FTP operation configuration example

Network requirements

As shown in Figure 36, 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 36 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-test1-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:

<table>
<thead>
<tr>
<th>Index</th>
<th>Response</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>173</td>
<td>Succeeded</td>
<td>2011-11-22 10:07:28.6</td>
</tr>
</tbody>
</table>

The output shows that Device A uses 173 milliseconds to upload a file to the FTP server.

HTTP operation configuration example

Network requirements

As shown in Figure 37, configure an HTTP operation on the NQA client to test the time required to obtain data from the HTTP server.

Figure 37 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.

# 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.
UDP jitter operation configuration example

Network requirements

As shown in Figure 38, configure a UDP jitter operation to test the jitter, delay, and round-trip time between Device A and Device B.

**Figure 38 Network diagram**

![Network Diagram](image)

**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
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 39, configure an SNMP operation to test the time the NQA client uses to get a response from the SNMP agent.
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:
The output shows that Device A uses 50 milliseconds to receive a response from the SNMP agent.

TCP operation configuration example

Network requirements

As shown in Figure 40, configure a TCP operation to test the time required for Device A and Device B to establish a TCP connection.

Figure 40 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:

<table>
<thead>
<tr>
<th>Index</th>
<th>Response</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>Succeeded</td>
<td>2011-11-22 10:27:25.1</td>
</tr>
</tbody>
</table>

The output shows that Device A uses 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 41, configure a UDP echo operation to test the round-trip time between Device A and Device B. The destination port number is 8000.

Figure 41 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.
UDP tracert operation configuration example

Network requirements

As shown in Figure 42, configure a UDP tracert operation to determine the routing path from Device A to Device B.

![Network diagram](image)

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. Execute the `ip ttl-expires enable` command on the intermediate devices and execute the `ip unreachables enable` command on Device B.
4. Configure Device A:
   
   # Create a UDP tracert operation.
   <DeviceA> system-view
   [DeviceA] nqa entry admin test1

The output shows that the round-trip time between Device A and port 8000 on Device B is 25 milliseconds.
[DeviceA-nqa-admin-test1] type udp-tracert

# Specify 10.2.2.2 as the destination IP address.
[DeviceA-nqa-admin-test1-udp-tracert] destination ip 10.2.2.2

# Set the destination port number to 33434.
[DeviceA-nqa-admin-test1-udp-tracert] destination port 33434

# Configure Device A to perform three probes to each hop.
[DeviceA-nqa-admin-test1-udp-tracert] probe count 3

# Set the probe timeout time to 500 milliseconds.
[DeviceA-nqa-admin-test1-udp-tracert] probe timeout 500

# Configure the UDP tracert operation to repeat every 5000 milliseconds.
[DeviceA-nqa-admin-test1-udp-tracert] frequency 5000

# Specify VLAN-interface 2 as the output interface for UDP packets.
[DeviceA-nqa-admin-test1-udp-tracert] out interface vlan-interface 2

# Enable the no-fragmentation feature.
[DeviceA-nqa-admin-test1-udp-tracert] no-fragment enable

# Set the maximum number of consecutive probe failures to 6.
[DeviceA-nqa-admin-test1-udp-tracert] max-failure 6

# Set the TTL value to 1 for UDP packets in the start round of the UDP tracert operation.
[DeviceA-nqa-admin-test1-udp-tracert] init-ttl 1

# Start the UDP tracert operation.
[DeviceA] nqa schedule admin test1 start-time now lifetime forever

# After the UDP tracert operation runs for a period of time, stop the operation.
[DeviceA] undo nqa schedule admin test1

# Display the most recent result of the UDP tracert operation.
[DeviceA] display nqa result admin test1

NQA entry (admin admin, tag test1) test results:
Send operation times: 6  Receive response times: 6
Min/Max/Average round trip time: 1/1/1
Square-Sum of round trip time: 1
Last succeeded probe time: 2013-09-09 14:46:06.2

Extended results:
Packet loss in test: 0%
Failures due to timeout: 0
Failures due to internal error: 0
Failures due to other errors: 0

UDP-tracert results:
<table>
<thead>
<tr>
<th>TTL</th>
<th>Hop IP</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.1.1.1</td>
<td>2013-09-09 14:46:03.2</td>
</tr>
<tr>
<td>2</td>
<td>10.2.2.2</td>
<td>2013-09-09 14:46:06.2</td>
</tr>
</tbody>
</table>

# Display the history records of the UDP tracert operation.
[DeviceA] display nqa history admin test1

NQA entry (admin admin, tag test1) history records:
<table>
<thead>
<tr>
<th>Index</th>
<th>TTL</th>
<th>Response</th>
<th>Hop IP</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>10.2.2.2</td>
<td>Succeeded</td>
<td>2013-09-09 14:46:06.2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>10.2.2.2</td>
<td>Succeeded</td>
<td>2013-09-09 14:46:05.2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>10.2.2.2</td>
<td>Succeeded</td>
<td>2013-09-09 14:46:04.2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3.1.1.1</td>
<td>Succeeded</td>
<td>2013-09-09 14:46:03.2</td>
</tr>
</tbody>
</table>
Voice operation configuration example

Network requirements

As shown in Figure 43, configure a voice operation to test jitters 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 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 44, configure a DLSw operation to test the response time of the DLSw device.

**Figure 44 Network diagram**

![Network diagram](image)

**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 45, configure a path jitter operation to test the round trip time and jitters from Device A to Device B and Device C.

Figure 45 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 46, 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 46 Network diagram**

![Network diagram](image)

**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] 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

# On Switch A, display information about all the track entries.

[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

# On Switch A, display information about all the track entries.

[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

<table>
<thead>
<tr>
<th>Destination/Mask</th>
<th>Proto</th>
<th>Pre</th>
<th>Cost</th>
<th>NextHop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0/32</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1</td>
<td>InLoop0</td>
</tr>
<tr>
<td>10.2.1.0/24</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>10.2.1.2</td>
<td>Vlan3</td>
</tr>
<tr>
<td>10.2.1.0/32</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>10.2.1.2</td>
<td>Vlan3</td>
</tr>
<tr>
<td>10.2.1.2/32</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1</td>
<td>InLoop0</td>
</tr>
<tr>
<td>10.2.1.255/32</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>10.2.1.2</td>
<td>Vlan3</td>
</tr>
<tr>
<td>127.0.0.0/8</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1</td>
<td>InLoop0</td>
</tr>
<tr>
<td>127.0.0.0/32</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1</td>
<td>InLoop0</td>
</tr>
<tr>
<td>127.0.0.1/32</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1</td>
<td>InLoop0</td>
</tr>
<tr>
<td>127.255.255.255/32</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1</td>
<td>InLoop0</td>
</tr>
<tr>
<td>224.0.0.0/4</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>NULL0</td>
</tr>
<tr>
<td>224.0.0.0/24</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>NULL0</td>
</tr>
<tr>
<td>255.255.255.255/32</td>
<td>Direct</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1</td>
<td>InLoop0</td>
</tr>
</tbody>
</table>

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 47, configure an ICMP template for a feature to perform the ICMP echo operation from Device A to Device B.

**Figure 47 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
# Configure the NQA client to notify the feature of the successful operation event if the number of consecutive successful probes reaches 2.
[DeviceA-nqatplt-icmp-icmp] reaction trigger probe-pass 2
# Configure the NQA client to notify the feature of the operation failure if the number of consecutive failed probes reaches 2.
[DeviceA-nqatplt-icmp-icmp] reaction trigger probe-fail 2

DNS template configuration example

Network requirements

As shown in Figure 48, 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 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 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
# Configure the NQA client to notify the feature of the successful operation event if the number of consecutive successful probes reaches 2.
[DeviceA-nqatplt-dns-dns] reaction trigger probe-pass 2
# Configure the NQA client to notify the feature of the operation failure if the number of consecutive failed probes reaches 2.

[DeviceA-nqatplt-dns-dns] reaction trigger probe-fail 2

## TCP template configuration example

### Network requirements

As shown in Figure 49, 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 49 Network diagram**

![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
   - # Configure the NQA client to notify the feature of the successful operation event if the number of consecutive successful probes reaches 2.
     `[DeviceA-nqatplt-tcp-tcp] reaction trigger probe-pass 2
   - # Configure the NQA client to notify the feature of the operation failure if the number of consecutive failed probes reaches 2.
     `[DeviceA-nqatplt-tcp-tcp] reaction trigger probe-fail 2

## UDP template configuration example

### Network requirements

As shown in Figure 50, configure a UDP template for a feature to perform the UDP operation. The operation tests whether Device A can receive a response from Device B.
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 UDP port 9000.
   [DeviceB] nqa server udp-echo 10.2.2.2 9000
4. Configure Device A:
   # Create UDP template udp.
   <DeviceA> system-view
   [DeviceA] nqa template udp udp
   # Specify 10.2.2.2 as the destination IP address.
   [DeviceA-nqatplt-udp-udp] destination ip 10.2.2.2
   # Set the destination port number to 9000.
   [DeviceA-nqatplt-udp-udp] destination port 9000
   # Configure the NQA client to notify the feature of the successful operation event if the number of consecutive successful probes reaches 2.
   [DeviceA-nqatplt-udp-udp] reaction trigger probe-pass 2
   # Configure the NQA client to notify the feature of the operation failure if the number of consecutive failed probes reaches 2.
   [DeviceA-nqatplt-udp-udp] reaction trigger probe-fail 2

HTTP template configuration example

Network requirements

As shown in Figure 51, 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.

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.

# Configure the HTTP operation to get data from the HTTP server.
[DeviceA-nqatplt-http-http] operation get

# Configure the NQA client to notify the feature of the successful operation event if the number of consecutive successful probes reaches 2.

# Configure the NQA client to notify the feature of the operation failure if the number of consecutive failed probes reaches 2.

### FTP template configuration example

#### Network requirements

As shown in Figure 52, 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 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 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.

# 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.
# Configure the NQA client to notify the feature of the successful operation event if the number of consecutive successful probes reaches 2.

```console
```

# Configure the NQA client to notify the feature of the operation failure if the number of consecutive failed probes reaches 2.

```console
```
Configuring port mirroring

The port mirroring feature is available on Layer 2 Ethernet interfaces.

Overview

Port mirroring refers to the process of copying the packets passing through a port to the monitor port connecting to a monitoring device for packet analysis.

Terminology

The following terms are used in port mirroring configuration.

**Mirroring source**

The mirroring source can be one or more monitored ports, which are called "source ports." Packets passing through them are copied to a port connecting to a monitoring device for packet analysis. (The copies are called "mirrored packets.") The device where the mirroring source resides is called a "source device."

**Mirroring destination**

The mirroring destination is the destination port, also known as the monitor port, of mirrored packets and connects to the monitoring device. The device where the monitor port resides is called the "destination device." Mirrored packets are sent out of the monitor port to the monitoring device.

A monitor port might receive multiple duplicates of a packet in some networks because it can monitor multiple mirroring sources. For example, assume that Port 1 is monitoring bidirectional traffic on Port 2 and Port 3 on the same device. If a packet travels from Port 2 to Port 3, two duplicates of the packet are received on Port 1.

**Mirroring direction**

The mirroring direction specifies that the inbound, outbound, or bidirectional traffic can be copied on a mirroring source.

- **Inbound**—Copies packets received on a mirroring source.
- **Outbound**—Copies packets sent out of a mirroring source.
- **Bidirectional**—Copies packets received and sent on a mirroring source.

**NOTE:**

- For inbound traffic mirroring, the VLAN tag in the original packet is copied to the mirrored packet.
- For outbound traffic mirroring, the VLAN tag in the mirrored packet identifies the VLAN to which the packet belongs before it is sent out of the source port.

**Mirroring group**

Port mirroring is implemented through mirroring groups, which include local, remote source, and remote destination mirroring groups. For more information about the mirroring groups, see “Port mirroring classification and implementation.”

**Reflector port, egress port, and remote probe VLAN**

A reflector port, remote probe VLAN, and an egress port 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 source and the mirroring destination are on the same device.
- **Remote port mirroring** — The mirroring source and the mirroring destination are on different devices.

### Local port mirroring

In local port mirroring:

- The source device is directly connected to the data monitoring device.
- The source device can act as the destination device to forward mirrored packets to the data monitoring device.

A mirroring group that contains the mirroring source and the mirroring destination on the device is called a "local mirroring group."

**Figure 53 Local port mirroring implementation**

As shown in Figure 53, the source port GigabitEthernet 1/0/1 and monitor port GigabitEthernet 1/0/2 reside on the same device. Packets received on GigabitEthernet 1/0/1 are copied to GigabitEthernet 1/0/2, which then forwards the packets to the data monitoring device for analysis.

### Remote port mirroring

In remote port mirroring:

- The source device is not directly connected to the data monitoring device.
- The source device copies mirrored packets to the destination device, which forwards them to the data monitoring device.

The mirroring source and the mirroring destination reside on different devices and in different mirroring groups. The mirroring group that contains the mirroring source or the mirroring destination is called a "remote source group" or "remote destination group," respectively. The devices between the source devices and destination device are intermediate devices.
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 broadcast the packets in the remote probe VLAN and transmit the packets to the destination device. Upon receiving the mirrored packets, the destination device checks whether their VLAN IDs are the same as the remote probe VLAN ID. If yes, the device forwards them to the data monitoring device through the monitor port.

Figure 54 Layer 2 remote port mirroring implementation

To make sure 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, you must 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

Local port mirroring takes effect only when the source ports, and the monitor port are configured.

Local port mirroring configuration task list

<table>
<thead>
<tr>
<th>Tasks at a glance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (Required.) Creating a local mirroring group</td>
</tr>
<tr>
<td>2. (Required.) Configuring source ports for the local mirroring group</td>
</tr>
<tr>
<td>3. (Required.) Configuring the monitor port for the local mirroring group</td>
</tr>
</tbody>
</table>
Creating a local mirroring group

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create a local mirroring group.</td>
<td>mirroring-group group-id local</td>
</tr>
</tbody>
</table>

Configuring source ports for the local mirroring group

You can configure a list of source ports for a mirroring group in system view. Also, you can assign a port to a 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 maximum of two mirroring resources can be assigned to a source port. A port can be assigned to a maximum of two mirroring groups as a unidirectional source port, inbound in one group and outbound in the other. A port can be assigned to only one mirroring group as a bidirectional source port.
- You can configure only one mirroring group to monitor the outbound or bidirectional traffic of source ports.
- If a port is configured as a source port, the port cannot be used as a reflector port, egress port, or monitor port.

Configuring source ports in system view

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Configure source ports for the specified local mirroring group.</td>
<td>mirroring-group group-id mirroring-port interface-list { both</td>
</tr>
</tbody>
</table>

Configuring source ports in interface view

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td>interface interface-type interface-number</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the port as a source port for the specified local mirroring group.</td>
<td>mirroring-group group-id mirroring-port { both</td>
</tr>
</tbody>
</table>
Configuring the monitor port for the local mirroring group

You can configure the monitor port for a mirroring group in system view. Also, you can assign a port to a mirroring group as the monitor port in interface view. The two methods have the same result.

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 for port mirroring. This makes the data monitoring device receive only the mirrored traffic rather than a mix of mirrored traffic and correctly forwarded traffic.

Configuring the monitor port in system view

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Configure the monitor port for the specified local mirroring group.</td>
<td>By default, no monitor port is configured for a local mirroring group.</td>
</tr>
</tbody>
</table>

Configuring the monitor port in interface view

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td>N/A</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the port as the monitor port for the specified mirroring group.</td>
<td>By default, a port does not act as the monitor port for any local mirroring group.</td>
</tr>
</tbody>
</table>

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. As a best practice, 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, you must 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>mirroring-group group-id remote-source</td>
<td>By default, no mirroring groups exist on a device.</td>
</tr>
<tr>
<td>3.</td>
<td>mirroring-group group-id mirroring-port mirroring-port-list { both</td>
<td>inbound</td>
</tr>
<tr>
<td>4.</td>
<td>mirroring-group group-id reflector-port reflector-port</td>
<td>By default, no reflector port is configured for a mirroring group.</td>
</tr>
<tr>
<td>5.</td>
<td>vlan vlan-id</td>
<td>By default, only VLAN 1 (system default VLAN) exists.</td>
</tr>
<tr>
<td>6.</td>
<td>port interface-list</td>
<td>By default, all ports are in VLAN 1.</td>
</tr>
<tr>
<td>7.</td>
<td>quit</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Configuring Layer 2 remote port mirroring

Configuring Layer 2 remote port mirroring is to configure remote mirroring groups. When doing this, configure a remote source group on the source device and a cooperating remote destination group on the destination device. If intermediate devices exist, configure the intermediate devices to allow the remote probe VLAN to pass through.

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. If MVRP is enabled, MVRP might register the remote probe VLAN to unexpected ports, resulting in undesired duplicates. For more information about MVRP, see Layer 2—LAN Switching Configuration Guide.
- As a best practice, configure the destination device first, then the intermediate devices, and then the source device.

<table>
<thead>
<tr>
<th>Tasks at a glance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Required.) Configuring a remote destination group on the destination device:</td>
</tr>
<tr>
<td>1. Creating a remote destination group</td>
</tr>
<tr>
<td>2. Configuring the monitor port for a remote destination group</td>
</tr>
<tr>
<td>3. Configuring the remote probe VLAN for a remote destination group</td>
</tr>
<tr>
<td>4. Assigning the monitor port to the remote probe VLAN</td>
</tr>
<tr>
<td>(Required.) Configuring a remote source group on the source device:</td>
</tr>
<tr>
<td>1. Creating a remote source group</td>
</tr>
<tr>
<td>2. Configuring source ports for a remote source group</td>
</tr>
<tr>
<td>3. Configuring the egress port for a remote source group</td>
</tr>
<tr>
<td>4. Configuring the remote probe VLAN for a remote source group</td>
</tr>
</tbody>
</table>

Configuring a remote destination group on the destination device

To configure a remote destination group, perform the following tasks on the destination device.

Creating a remote destination group

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>mirroring-group group-id remote-destination</td>
<td>By default, no remote destination group exists on a device.</td>
</tr>
</tbody>
</table>
Configuring the monitor port for a remote destination group

You can configure the monitor port for a mirroring group in system view. Also, you can assign a port to a mirroring group as the monitor port in interface view. The two methods have the same results.

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. This makes the data monitoring device receive only the mirrored traffic rather than a mix of mirrored traffic and correctly forwarded traffic.
- A mirroring group must contain only one monitor port, and a monitor port can belong to only one mirroring group.

To configure the monitor port for a remote destination group in system view:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Configure the monitor port for the specified</td>
<td>By default, no monitor port is configured for a remote destination group.</td>
</tr>
<tr>
<td></td>
<td>remote destination group.</td>
<td></td>
</tr>
</tbody>
</table>

To configure the monitor port for a remote destination group in interface view:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td>N/A</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the port as the monitor port for</td>
<td>By default, a port does not act as the monitor port for any remote</td>
</tr>
<tr>
<td></td>
<td>the specified remote destination group.</td>
<td>destination group.</td>
</tr>
</tbody>
</table>

Configuring the remote probe VLAN for a remote destination group

You must first create a static VLAN before you configure it as a remote probe VLAN.

When you configure the remote probe VLAN for a remote destination group, follow these restrictions and guidelines:

- 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 destination groups on the source and destination devices.

To configure the remote probe VLAN for a remote destination group:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Configure the remote probe VLAN for the specified remote destination group.</td>
<td>By default, no remote probe VLAN is configured for a remote destination group.</td>
</tr>
</tbody>
</table>
Assigning the monitor port to the remote probe VLAN

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
</tr>
<tr>
<td>2.</td>
<td>Enter the interface view of the monitor port.</td>
</tr>
</tbody>
</table>
| 3. | Assign the port to the 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create a remote source group.</td>
<td>mirroring-group group-id remote-source</td>
</tr>
</tbody>
</table>

By default, no remote source group exists on a device.

Configuring source ports for a remote source group

You can configure a list of source ports for a mirroring group in system view. Also, you can assign a port to a mirroring group as a source port in interface view. To assign multiple ports to a mirroring 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 to the remote probe VLAN.
- A mirroring group can contain multiple source ports.
- A maximum of two mirroring resources can be assigned to a source port. A port can be assigned to a maximum of two mirroring groups as a unidirectional source port, inbound in one group and outbound in the other. A port can be assigned to only one mirroring group as a bidirectional source port.
- You can configure only one mirroring group to monitor the outbound or bidirectional traffic of source ports.
- If a port is configured as a source port, the port cannot be used as a reflector port, monitor port, or an egress port.

To configure source ports for a remote source group in system view:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Configure source ports for the specified remote source group.</td>
<td>mirroring-group group-id mirroring-port interface-list { both</td>
</tr>
</tbody>
</table>

By default, no source port is configured for a remote source group.
To configure a source port for a remote source group in interface view:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td>interface interface-type interface-number</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the port as a source port for the specified remote source group.</td>
<td>mirroring-group group-id mirroring-port { both</td>
</tr>
</tbody>
</table>

**Configuring the egress port for a remote source group**

You can configure the egress port for a mirroring group in system view. Also, you can assign a port to a mirroring group as the egress port in interface view. The two configuration methods have the same result.

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.

To configure the egress port for a remote source group in system view:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Configure the egress port for the specified remote source group.</td>
<td>mirroring-group group-id monitor-egress interface-type interface-number</td>
</tr>
</tbody>
</table>

To configure the egress port for a remote source group in interface view:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter interface view.</td>
<td>interface interface-type interface-number</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the port as the egress port for the specified remote source group.</td>
<td>mirroring-group group-id monitor-egress</td>
</tr>
</tbody>
</table>

**Configuring the remote probe VLAN for a remote source group**

You must first create a static VLAN before you configure it as a remote probe VLAN.
When you configure the remote probe VLAN for a remote source group, follow these restrictions and guidelines:

- When a VLAN is configured as a remote probe VLAN, use the remote probe 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Configure the remote probe VLAN for the specified remote source group.</td>
<td>mirroring-group group-id remote-probe vlan vlan-id</td>
<td>By default, no remote probe VLAN is configured for a remote source group.</td>
</tr>
</tbody>
</table>

Displaying and maintaining port mirroring

Execute **display** commands in any view.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display mirroring group information.</td>
<td>display mirroring-group { group-id</td>
</tr>
</tbody>
</table>

Port mirroring configuration examples

Local port mirroring configuration example

Network requirements

As shown in Figure 55, configure local port mirroring so the server can monitor the bidirectional traffic of Marketing Department and Technical Department.

Figure 55 Network diagram
Configuration procedure

# Create local mirroring group 1.
<Device> system-view
[Device] mirroring-group 1 local

# Configure GigabitEthernet 1/0/1 and GigabitEthernet 1/0/2 as source ports, and GigabitEthernet 1/0/3 as the monitor port for local mirroring group 1.
[Device] mirroring-group 1 mirroring-port gigabitethernet 1/0/1 gigabitethernet 1/0/2 both
[Device] mirroring-group 1 monitor-port gigabitethernet 1/0/3

# Disable the spanning tree feature on the monitor port GigabitEthernet 1/0/3.
[Device] interface gigabitethernet 1/0/3
[Device-GigabitEthernet1/0/3] undo stp enable
[Device-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:
        GigabitEthernet1/0/1  Both
        GigabitEthernet1/0/2  Both
    Monitor port: GigabitEthernet1/0/3

The output shows that you can monitor all packets received and sent by Marketing Department and Technical Department on the server.

Layer 2 remote port mirroring configuration example

Network requirements

As shown in Figure 56, configure Layer 2 remote port mirroring so the server can monitor the bidirectional traffic of Marketing Department.

Figure 56 Network diagram
Configuration procedure

1. Configure Device C (the destination device):
   # Configure GigabitEthernet 1/0/1 as a trunk port to permit the packets from VLAN 2 to pass through.
   <DeviceC> system-view
   [DeviceC] interface gigabitethernet 1/0/1
   [DeviceC-GigabitEthernet1/0/1] port link-type trunk
   [DeviceC-GigabitEthernet1/0/1] port trunk permit vlan 2
   [DeviceC-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 GigabitEthernet 1/0/2 as the monitor port of the mirroring group.
   [DeviceC] mirroring-group 2 remote-probe vlan 2
   [DeviceC] interface gigabitethernet 1/0/2
   [DeviceC-GigabitEthernet1/0/2] mirroring-group 2 monitor-port
   # Disable the spanning tree feature on GigabitEthernet 1/0/2.
   [DeviceC-GigabitEthernet1/0/2] undo stp enable
   # Assign GigabitEthernet 1/0/2 to VLAN 2 as an access port.
   [DeviceC-GigabitEthernet1/0/2] port access vlan 2
   [DeviceC-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 GigabitEthernet 1/0/1 as a trunk port to permit the packets from VLAN 2 to pass through.
   [DeviceB] interface gigabitethernet 1/0/1
   [DeviceB-GigabitEthernet1/0/1] port link-type trunk
   [DeviceB-GigabitEthernet1/0/1] port trunk permit vlan 2
   [DeviceB-GigabitEthernet1/0/1] quit
   # Configure GigabitEthernet 1/0/2 as a trunk port to permit the packets from VLAN 2 to pass through.
   [DeviceB] interface gigabitethernet 1/0/2
   [DeviceB-GigabitEthernet1/0/2] port link-type trunk
   [DeviceB-GigabitEthernet1/0/2] port trunk permit vlan 2
   [DeviceB-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 GigabitEthernet 1/0/1 as a source port and GigabitEthernet 1/0/2 as the egress port in the mirroring group.
[DeviceA] mirroring-group 1 mirroring-port gigabitethernet 1/0/1 both
[DeviceA] mirroring-group 1 monitor-egress gigabitethernet 1/0/2

# Configure GigabitEthernet 1/0/2 as a trunk port to permit the packets from VLAN 2 to pass through.
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-type trunk
[DeviceA-GigabitEthernet1/0/2] port trunk permit vlan 2

# Disable the spanning tree feature on GigabitEthernet 1/0/2.
[DeviceA-GigabitEthernet1/0/2] undo stp enable
[DeviceA-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: 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:
        GigabitEthernet1/0/1 Both
    Monitor egress port: Gigabitethernet1/0/2
    Remote probe VLAN: 2

The output shows that you can monitor all packets received and sent by Marketing Department on the server.

Local port mirroring with multiple monitor ports configuration example

Network requirements

As shown in Figure 57, configure port mirroring so servers A, B, and C can monitor the bidirectional traffic of the three departments.
Figure 57 Network diagram

**Configuration procedure**

# Create remote source group 1.

```yaml
<DeviceA> system-view
[DeviceA] mirroring-group 1 remote-source
```

# Configure GigabitEthernet 1/0/1 through GigabitEthernet 1/0/3 as source ports of the remote source group.

```yaml
[DeviceA] mirroring-group 1 mirroring-port gigabitethernet 1/0/1 to gigabitethernet 1/0/3 both
```

# Configure an unused port (GigabitEthernet 1/0/5, for example) of Device A as the reflector port of the remote source group.

```yaml
[DeviceA] mirroring-group 1 reflector-port gigabitethernet 1/0/5
```

This operation may delete all settings made on the interface. Continue? [Y/N]: y

# Create VLAN 10, and assign ports GigabitEthernet 1/0/11 through GigabitEthernet 1/0/13 to VLAN 10.

```yaml
[DeviceA] vlan 10
[DeviceA-vlan10] port gigabitethernet 1/0/11 to gigabitethernet 1/0/13
[DeviceA-vlan10] quit
```

# Configure VLAN 10 as the remote probe VLAN of the remote source group.

```yaml
[DeviceA] mirroring-group 1 remote-probe vlan 10
```
Configuring traffic mirroring

The traffic mirroring feature is available on Layer 2 Ethernet interfaces.

Traffic mirroring copies the specified packets to the specified destination for packet analyzing and monitoring. It is implemented through QoS policies. To configure traffic mirroring, perform the following tasks:

- Define traffic classes and configure match criteria to classify packets to be mirrored
- Configure traffic behaviors to mirror packets that fit the match criteria to the specified destination.

Traffic mirroring allows you to flexibly classify packets to be analyzed by defining match criteria. For more information about QoS policies, traffic classes, and traffic behaviors, see *ACL and QoS Configuration Guide*.

You can configure the traffic to be mirrored to the following destinations:

- **Interface**—Mirroring traffic to an interface copies the matching packets to an interface connecting to a data monitoring device. The data monitoring device analyzes the packets received on the interface.
- **CPU**—Mirroring traffic to a CPU copies the matching packets to a CPU, whose IRF member device is configured with traffic mirroring. The CPU analyzes the packets or delivers them to upper layers.

Traffic mirroring configuration task list

<table>
<thead>
<tr>
<th>Tasks at a glance</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Required.) Configuring match criteria</td>
<td>N/A</td>
</tr>
<tr>
<td>(Required.) Configuring a traffic behavior</td>
<td></td>
</tr>
<tr>
<td>(Required.) Configuring a QoS policy</td>
<td></td>
</tr>
<tr>
<td>(Required.) Applying a QoS policy:</td>
<td></td>
</tr>
<tr>
<td>- Applying a QoS policy to an interface</td>
<td></td>
</tr>
<tr>
<td>- Applying a QoS policy to a VLAN</td>
<td></td>
</tr>
<tr>
<td>- Applying a QoS policy globally</td>
<td></td>
</tr>
<tr>
<td>- Applying a QoS policy to the control plane</td>
<td></td>
</tr>
</tbody>
</table>

For more information about the following commands except the `mirror-to` command, see *ACL and QoS Command Reference*.

Configuring match criteria

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td><code>system-view</code></td>
<td>N/A</td>
</tr>
<tr>
<td>2. Create a class and enter class view.</td>
<td>`traffic classifier tcl-name [ operator { and</td>
<td>or } ]`</td>
</tr>
<tr>
<td>3. Configure match criteria.</td>
<td><code>if-match match-criteria</code></td>
<td>By default, no match criterion is configured in a traffic class.</td>
</tr>
</tbody>
</table>
Configuring a traffic behavior

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create a traffic behavior and enter traffic behavior view.</td>
<td>traffic behavior behavior-name</td>
</tr>
<tr>
<td>3.</td>
<td>Specify a mirroring destination for the traffic behavior.</td>
<td>Mirror traffic to an interface: mirror-to interface interface-type interface-number • Mirror traffic to a CPU: mirror-to cpu</td>
</tr>
</tbody>
</table>

NOTE:
After you configure a traffic behavior, you can use the display traffic behavior command in any view to display traffic behavior configuration.

Configuring a QoS policy

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create a QoS policy and enter the QoS policy view.</td>
<td>qos policy policy-name</td>
</tr>
<tr>
<td>3.</td>
<td>Associate a class with a traffic behavior in the QoS policy.</td>
<td>classifier tcl-name behavior behavior-name</td>
</tr>
</tbody>
</table>

NOTE:
After the preceding configuration, you can use the display qos policy command in any view to display QoS policy configuration.

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 a 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.

The device does not support mirroring outbound traffic of an aggregate interface.

To apply a QoS policy to an interface:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
</tr>
</tbody>
</table>
Applying a QoS policy to a VLAN

You can apply a QoS policy to a VLAN to mirror the traffic in a specified direction on all ports in the VLAN.

To apply the QoS policy to a VLAN:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>qos vlan-policy policy-name vlan vlan-id-list { inbound</td>
</tr>
</tbody>
</table>

Applying a QoS policy globally

You can apply a QoS policy globally to mirror the traffic in a specified direction on all ports.

To apply a QoS policy globally:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>qos apply policy policy-name global { inbound</td>
</tr>
</tbody>
</table>

Applying a QoS policy to the control plane

You can apply a QoS policy to the control plane to mirror the traffic in a specified direction on all ports of the control plane.

To apply a QoS policy to the control plane:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>control-plane slot slot-number</td>
</tr>
<tr>
<td>3.</td>
<td>qos apply policy policy-name inbound</td>
</tr>
</tbody>
</table>
Traffic mirroring configuration example

Network requirements

As shown in Figure 58, different departments of a company use IP addresses on different subnets. Marketing Department and Technical Department use the IP addresses on subnets 192.168.1.0/24 and 192.168.2.0/24, respectively. The working hour of the company is from 8:00 to 18:00 on weekdays.

Configure traffic mirroring so that the server can monitor following traffic:

- All traffic that Technical Department sends to access the Internet.
- IP traffic that Technical Department sends to Marketing Department during working hours.

Figure 58 Network diagram

Configuration procedure

# Create a working hour range named 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 Technical Department to access the Internet and to 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 GigabitEthernet 1/0/3.

```
[DeviceA] traffic behavior tech_b
[DeviceA-behavior-tech_b] mirror-to interface 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 GigabitEthernet 1/0/4.

```
[DeviceA] interface gigabitethernet 1/0/4
[DeviceA-GigabitEthernet1/0/4] qos apply policy tech_p inbound
[DeviceA-GigabitEthernet1/0/4] quit
```

Verifying the configuration

# Verify that you can monitor the following traffic through the server:
- All traffic sent by Technical Department to access the Internet.
- The IP traffic that Technical Department sends to Marketing Department during working hours.

(Details not shown.)
Configuring sFlow

Sampled Flow (sFlow) is a traffic monitoring technology.

As shown in Figure 59, 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 59 sFlow system

Protocols and standards

- sFlow.org, *sFlow Version 5*

sFlow configuration task list

<table>
<thead>
<tr>
<th>Tasks at a glance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Required.) <strong>Configuring the sFlow agent and sFlow collector information</strong></td>
</tr>
<tr>
<td>Perform at least one of the following tasks:</td>
</tr>
<tr>
<td>• <strong>Configuring flow sampling</strong></td>
</tr>
<tr>
<td>• <strong>Configuring counter sampling</strong></td>
</tr>
</tbody>
</table>
Configuring the sFlow agent and sFlow collector information

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td></td>
<td>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. <strong>NOTE:</strong> • As a best practice, 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.</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>(Optional.) Configure an IP address for the sFlow agent.</td>
<td>sflow agent { ip ip-address</td>
</tr>
<tr>
<td></td>
<td>By default, no sFlow collector information is configured.</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Configure the sFlow collector information.</td>
<td>sflow collector collector-id { ip ip-address</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Specify the source IP address of sFlow packets.</td>
<td>sflow source { ip ip-address</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter Layer 2 Ethernet interface view.</td>
<td>interface interface-type interface-number</td>
</tr>
<tr>
<td>3.</td>
<td>(Optional.) Set the flow sampling mode.</td>
<td>sflow sampling-mode { determine</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>interface interface-type interface-number</td>
<td>N/A</td>
</tr>
<tr>
<td>3.</td>
<td>sflow counter interval interval-time</td>
<td>By default, counter sampling is disabled.</td>
</tr>
<tr>
<td>4.</td>
<td>sflow counter collector collector-id</td>
<td>By default, no sFlow collector is specified for counter sampling.</td>
</tr>
</tbody>
</table>

Displaying and maintaining sFlow

Execute **display** commands in any view.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display sFlow configuration.</td>
<td>display sflow</td>
</tr>
</tbody>
</table>

sFlow configuration example

Network requirements

As shown in Figure 60, perform the following tasks:
• Configure flow sampling in random mode and counter sampling on GigabitEthernet 1/0/1 of the device to monitor traffic on the port.
• Configure the device to send sampled information in sFlow packets through GigabitEthernet 1/0/3 to the sFlow collector.

Figure 60 Network diagram

Configuration procedure

1. Configure the IP addresses and subnet masks for interfaces, as shown in Figure 60. (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 GigabitEthernet 1/0/1.
   [Device] interface gigabitethernet 1/0/1
   [Device-GigabitEthernet1/0/1] sflow counter interval 120
   # Specify sFlow collector 1 for counter sampling.
   [Device-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-GigabitEthernet1/0/1] sflow sampling-mode random
   [Device-GigabitEthernet1/0/1] sflow sampling-rate 4000
   # Specify sFlow collector 1 for flow sampling.
   [Device-GigabitEthernet1/0/1] sflow flow collector 1

Verifying the configuration

# Verify that GigabitEthernet 1/0/1 enabled with sFlow is active, and sFlow is operating correctly.
[Device-GigabitEthernet1/0/1] display sflow
sFlow datagram version: 5
Global information:
Agent IP: 3.3.3.1 (CLI)
Source address:
Collector information:
<table>
<thead>
<tr>
<th>ID</th>
<th>IP</th>
<th>Port</th>
<th>Aging</th>
<th>Size</th>
<th>VPN-instance</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.3.3.2</td>
<td>6343</td>
<td>N/A</td>
<td>1400</td>
<td></td>
<td>netserver</td>
</tr>
</tbody>
</table>
Port information:
<table>
<thead>
<tr>
<th>Interface</th>
<th>CID</th>
<th>Interval(s)</th>
<th>FID</th>
<th>MaxHLen</th>
<th>Rate</th>
<th>Mode</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE1/0/1</td>
<td>1</td>
<td>120</td>
<td>1</td>
<td>128</td>
<td>4000</td>
<td>Random</td>
<td>Active</td>
</tr>
</tbody>
</table>

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 (as a best practice, use the default) that flow sampling can copy per packet.
Monitoring and maintaining processes

H3C Comware 7 is a full-featured, modular, and scalable network operating system based on the Linux kernel. Comware 7 software features run the following types of independent processes:

- **User process**—Runs in user space. Most Comware 7 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 7 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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display memory usage.</td>
<td>display memory [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
<tr>
<td>Display process state information.</td>
<td>display process [ all</td>
</tr>
<tr>
<td>Display CPU usage for all processes.</td>
<td>display process cpu [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
<tr>
<td>Monitor process running state.</td>
<td>monitor process [ dumbtty ] [ iteration number ] [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
<tr>
<td>Monitor thread running state.</td>
<td>monitor thread [ dumbtty ] [ iteration number ] [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
</tbody>
</table>

For more 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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display log information for all user processes.</td>
<td>display process log [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
<tr>
<td>Display memory usage for all user processes.</td>
<td>display process memory [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
<tr>
<td>Display heap memory usage for a user process.</td>
<td>display process memory heap job job-id [ verbose ] [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
</tbody>
</table>
### 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, this feature determines that a deadloop has occurred and generates a deadloop message. In release 3113P05, this feature also causes the device to reboot to remove the deadloop.

To configure kernel thread deadloop detection:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enable kernel thread deadloop detection.</td>
<td>monitor kernel deadloop enable [ slot slot-number [ cpu cpu-number ] ]</td>
<td>In release 3113P05, this feature is disabled by default. In release 3116 and later, this feature is enabled by default.</td>
</tr>
</tbody>
</table>
3. (Optional.) Set the interval for identifying a kernel thread deadloop.

```
monitor kernel deadloop time interval [ slot slot-number [ cpu cpu-number ] ]
```

- In release 3113P05, the default is 8 seconds.
- In release 3116 and later, the default is 20 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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable kernel thread starvation detection.</td>
<td>monitor kernel starvation enable [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
<tr>
<td>3.</td>
<td>(Optional.) Set the interval for identifying a kernel thread starvation.</td>
<td>monitor kernel starvation time interval [ slot slot-number [ cpu cpu-number ] ]</td>
</tr>
</tbody>
</table>

### Displaying and maintaining kernel threads

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display kernel thread deadloop information.</td>
<td><code>display kernel deadloop show-number [ offset ] [ verbose ] [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Display kernel thread deadloop detection configuration.</td>
<td><code>display kernel deadloop configuration [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Display kernel thread exception information.</td>
<td><code>display kernel exception show-number [ offset ] [ verbose ] [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Task</td>
<td>Command</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Display kernel thread reboot information.</td>
<td><code>display kernel reboot show-number [ offset ] [ verbose ] [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Display kernel thread starvation information.</td>
<td><code>display kernel starvation show-number [ offset ] [ verbose ] [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Display kernel thread starvation detection configuration.</td>
<td><code>display kernel starvation configuration [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Clear kernel thread deadloop information.</td>
<td><code>reset kernel deadloop [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Clear kernel thread exception information.</td>
<td><code>reset kernel exception [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Clear kernel thread reboot information.</td>
<td><code>reset kernel reboot [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
<tr>
<td>Clear kernel thread starvation information.</td>
<td><code>reset kernel starvation [ slot slot-number [ cpu cpu-number ] ]</code></td>
</tr>
</tbody>
</table>
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 61.

Figure 61 EAA framework

Event sources

Event sources are software or hardware modules that trigger events (see Figure 61). For example, the CLI module triggers an event when you enter a command, and the Syslog module (the information center) triggers an event when it receives a log message.

Event monitors

EAA creates one event monitor 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.
- At least one action.
- At least 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>
<thead>
<tr>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLI</td>
<td>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.</td>
</tr>
<tr>
<td>Syslog</td>
<td>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.</td>
</tr>
<tr>
<td>Process</td>
<td>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).</td>
</tr>
</tbody>
</table>
| 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. |
| 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**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any event:</td>
<td></td>
</tr>
<tr>
<td>_event_id</td>
<td>Event ID.</td>
</tr>
<tr>
<td>_event_type</td>
<td>Event type.</td>
</tr>
<tr>
<td>_event_type_string</td>
<td>Event type description.</td>
</tr>
<tr>
<td>_event_time</td>
<td>Time when the event occurs.</td>
</tr>
<tr>
<td>_event_severity</td>
<td>Severity level of an event.</td>
</tr>
<tr>
<td>CLI:</td>
<td></td>
</tr>
</tbody>
</table>
### 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:

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

2. Configure a user-defined EAA environment variable.
   - **Command**: `rtm environment env-name env-value`  
   - **Remarks**: 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter CLI-defined policy view.</td>
<td>rtm cli-policy policy-name</td>
</tr>
<tr>
<td>3.</td>
<td>Configure an event in the policy.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Configure a CLI event:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>event cli { async [ skip ]</td>
<td>sync }</td>
</tr>
<tr>
<td></td>
<td>mode { execute</td>
<td>help</td>
</tr>
<tr>
<td></td>
<td>• Configure a hotplug event:</td>
<td>event hotplug [ slot slot-number ]</td>
</tr>
<tr>
<td></td>
<td>• Configure an interface event:</td>
<td>event interface interface-type interface-number monitor-obj monitor-obj start-op start-op start-val start-val restart-op restart-op restart-op restart-val restart-val interval interval</td>
</tr>
<tr>
<td></td>
<td>• Configure a process event:</td>
<td>event process { exception</td>
</tr>
<tr>
<td></td>
<td>• Configure an SNMP event:</td>
<td>event snmp oid oid monitor-obj { get</td>
</tr>
<tr>
<td></td>
<td>• Configure an SNMP-Notification event:</td>
<td>event snmp-notification oid oid oid-val oid-val op [ drop ]</td>
</tr>
<tr>
<td></td>
<td>• Configure a Syslog event:</td>
<td>event syslog priority level msg msg occurs times period period</td>
</tr>
<tr>
<td>4.</td>
<td>Configure the actions to take when the event occurs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Configure a CLI action:</td>
<td>action number cli command-line</td>
</tr>
<tr>
<td></td>
<td>• Configure a reboot action:</td>
<td>action number reboot [ slot slot-number ]</td>
</tr>
<tr>
<td></td>
<td>• Configure a logging action:</td>
<td>action number syslog priority level facility local-number msg msg</td>
</tr>
<tr>
<td></td>
<td>• Configure an active/standby switchover action:</td>
<td>action number switchover</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Remarks</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Assign a user role to the policy.</td>
<td><strong>user-role role-name</strong>&lt;br&gt;By default, a monitor policy contains user roles that its creator had at the time of policy creation.&lt;br&gt;A monitor policy supports a maximum of 64 valid user roles. After this limit is reached, you cannot assign additional roles to the policy.&lt;br&gt;An EAA policy cannot have both the <strong>security-audit</strong> user role and any other user roles. Any previously assigned user roles are automatically removed when you assign the <strong>security-audit</strong> user role to the policy. The <strong>security-audit</strong> user role takes effect. The previously assigned <strong>security-audit</strong> user role is automatically removed when you assign any other user roles to the policy. All user roles assigned after you assign the <strong>security-audit</strong> user role take effect.&lt;br&gt;For more information about user roles, see &quot;Fundamentals Configuration Guide.&quot;</td>
</tr>
<tr>
<td>6.</td>
<td>(Optional.) Configure the policy runtime.</td>
<td><strong>running-time time</strong>&lt;br&gt;The default runtime is 20 seconds.</td>
</tr>
<tr>
<td>7.</td>
<td>Enable the policy.</td>
<td><strong>commit</strong>&lt;br&gt;By default, CLI-defined policies are not enabled.&lt;br&gt;A CLI-defined policy can take effect only after you perform this step.</td>
</tr>
</tbody>
</table>

**Configuring a monitor policy by using Tcl**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Edit a Tcl script file (see Table 21).</td>
<td>N/A&lt;br&gt;The supported Tcl version is 8.5.8.</td>
</tr>
<tr>
<td>2.</td>
<td>Download the file to the device by using FTP or TFTP.</td>
<td>N/A&lt;br&gt;For more information about using FTP and TFTP, see Fundamentals Configuration Guide.</td>
</tr>
<tr>
<td>3.</td>
<td>Enter system view.</td>
<td><strong>system-view</strong>&lt;br&gt;N/A</td>
</tr>
</tbody>
</table>
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 first suspending these policies.

Write a Tcl script in two lines for a monitor policy, as shown in Table 21.

**Table 21 Tcl script requirements**

<table>
<thead>
<tr>
<th>Line</th>
<th>Content</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line 1</td>
<td>Event, user roles, and policy runtime</td>
<td>This line must take the following format: ::comware::rtm::event_register eventname arg1 arg2 arg3 ...user-role rolename1</td>
</tr>
<tr>
<td>Line 2</td>
<td>Actions</td>
<td>When you define an action, you can choose to specify a value or specify a variable name in $variable_name format for an argument. The following actions are available: • Standard Tcl commands. • EAA-specific Tcl commands. • Commands supported by the device.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>rtm scheduler suspend</td>
<td>To resume monitor policies, use the undo rtm scheduler suspend command.</td>
</tr>
</tbody>
</table>

**Displaying and maintaining EAA settings**

Execute display commands in any view.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display user-defined EAA environment variables.</td>
<td>display rtm environment [ var-name ]</td>
</tr>
</tbody>
</table>
# 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
<table>
<thead>
<tr>
<th>Type</th>
<th>Event</th>
<th>TimeRegistered</th>
<th>PolicyName</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLI</td>
<td>CLI</td>
<td>May 07 02:08:17 2013</td>
<td>test</td>
</tr>
</tbody>
</table>

# 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?
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
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

%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

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
%S 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>
<thead>
<tr>
<th>Category</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device information</td>
<td>Manufacturer</td>
</tr>
<tr>
<td></td>
<td>ManufacturerOUI</td>
</tr>
<tr>
<td></td>
<td>SerialNumber</td>
</tr>
<tr>
<td></td>
<td>HardwareVersion</td>
</tr>
<tr>
<td></td>
<td>SoftwareVersion</td>
</tr>
<tr>
<td>Operating status and information</td>
<td>DeviceStatus</td>
</tr>
<tr>
<td></td>
<td>UpTime</td>
</tr>
<tr>
<td>Configuration file</td>
<td>ConfigFile</td>
</tr>
<tr>
<td>CWMP settings</td>
<td>ACS URL</td>
</tr>
<tr>
<td></td>
<td>ACS username</td>
</tr>
<tr>
<td></td>
<td>ACS password</td>
</tr>
<tr>
<td></td>
<td>PeriodicInformEnable</td>
</tr>
<tr>
<td></td>
<td>PeriodicInformInterval</td>
</tr>
<tr>
<td></td>
<td>PeriodicInformTime</td>
</tr>
<tr>
<td></td>
<td>ConnectionRequestURL (CPE URL)</td>
</tr>
<tr>
<td></td>
<td>ConnectionRequestUsername (CPE username)</td>
</tr>
<tr>
<td></td>
<td>ConnectionRequestPassword (CPE password)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>RPC method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get</td>
<td>The ACS obtains the values of parameters on the CPE.</td>
</tr>
<tr>
<td>Set</td>
<td>The ACS modifies the values of parameters on the CPE.</td>
</tr>
</tbody>
</table>
| 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, as a best practice, 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.
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**

<table>
<thead>
<tr>
<th>Remarks</th>
<th>(Required.) Enabling CWMP from the CLI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tasks at a glance</strong></td>
<td><strong>Remarks</strong></td>
</tr>
<tr>
<td>(Required.) Configuring ACS attributes:</td>
<td>To use CWMP, you must enable CWMP from the CLI.</td>
</tr>
<tr>
<td>• (Required.) Configuring the preferred ACS attributes</td>
<td>The preferred ACS attributes are configurable from the CPE’s CLI, DHCP server, and ACS.</td>
</tr>
<tr>
<td>• Assigning ACS attributes from the DHCP server</td>
<td>The default ACS attributes are configurable only from the CLI.</td>
</tr>
<tr>
<td>• Configuring the preferred ACS attributes from the CLI</td>
<td></td>
</tr>
<tr>
<td>• (Optional.) Configuring the default ACS attributes from the CLI</td>
<td></td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view N/A</td>
</tr>
<tr>
<td>2.</td>
<td>Enter CWMP view.</td>
<td>cwmp</td>
</tr>
<tr>
<td>3.</td>
<td>Enable CWMP.</td>
<td>cwmp enable</td>
</tr>
</tbody>
</table>

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 H3C device, you can configure DHCP option 43 by using the **option 43 hex 01length 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`:

```plaintext
<Sysname> system-view
<Sysname> cwmp acs url http://169.254.76.31
<Sysname> cwmp acs username 1234
<Sysname> cwmp acs password 5678
```

Table 24 Hexadecimal forms of the ACS attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute value</th>
<th>Hexadecimal form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>39 characters</td>
<td>27</td>
</tr>
<tr>
<td>ACS URL</td>
<td><a href="http://169.254.76.31/acs">http://169.254.76.31/acs</a></td>
<td>687474703A2F2F3136392E3235342E37362E33313A373534372F61637320313233342035363738</td>
</tr>
<tr>
<td>ACS connect username</td>
<td>1234</td>
<td>3132333420</td>
</tr>
<tr>
<td>ACS connect password</td>
<td>5678</td>
<td>35363738</td>
</tr>
</tbody>
</table>

For more information about DHCP and DHCP Option 43, see *layer 3—IP Services Configuration Guide*.

**Configuring the preferred ACS attributes from the CLI**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>cwmp</td>
<td>N/A</td>
</tr>
<tr>
<td>3.</td>
<td>cwmp acs url url</td>
<td>By default, no preferred ACS URL has been configured.</td>
</tr>
<tr>
<td>4.</td>
<td>cwmp acs username username</td>
<td>By default, no username has been configured for authentication to the preferred ACS URL.</td>
</tr>
<tr>
<td>5.</td>
<td>cwmp acs password { cipher</td>
<td>simple } password</td>
</tr>
</tbody>
</table>
Configuring the default ACS attributes from the CLI

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter CWMP view.</td>
<td>cwmp</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the default ACS URL.</td>
<td>cwmp acs default url url</td>
</tr>
<tr>
<td>4.</td>
<td>Configure the username for authentication to the default ACS URL.</td>
<td>cwmp acs default username username</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Configure the password for authentication to the default ACS URL.</td>
<td>cwmp acs default password { cipher</td>
</tr>
</tbody>
</table>

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 can use only a username for authentication.

To configure ACS authentication parameters:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter CWMP view.</td>
<td>cwmp</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the username for authentication to the CPE.</td>
<td>cwmp cpe username username</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Configure the password for authentication to the CPE.</td>
<td>cwmp cpe password { cipher</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter CWMP view.</td>
<td>cwmp</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the provision code.</td>
<td>cwmp cpe provision-code provision-code</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter CWMP view.</td>
<td>cwmp</td>
</tr>
<tr>
<td>3.</td>
<td>Specify the interface that connects to the ACS as the CWMP connection interface.</td>
<td>cwmp cpe connect interface interface-type interface-number</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter CWMP view.</td>
<td>cwmp</td>
</tr>
<tr>
<td>3.</td>
<td>Enable the periodic Inform feature.</td>
<td>cwmp cpe inform interval enable</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Configure the Inform interval.</td>
<td>cwmp cpe inform interval seconds</td>
</tr>
</tbody>
</table>
Scheduling a connection initiation

To connect to the ACS for configuration or software update at a scheduled time:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enter CWMP view.</td>
<td>cwmp</td>
<td>N/A</td>
</tr>
<tr>
<td>3. Schedule a connection initiation.</td>
<td>cwmp cpe inform time time</td>
<td>By default, no connection initiation has been scheduled.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enter CWMP view.</td>
<td>cwmp</td>
<td>N/A</td>
</tr>
<tr>
<td>3. Configure the maximum number of connection retries.</td>
<td>cwmp cpe connect retry times</td>
<td>By default, the CPE retries a failed connection until the connection is established.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enter CWMP view.</td>
<td>cwmp</td>
<td>N/A</td>
</tr>
<tr>
<td>3. Set the close-wait timer.</td>
<td>cwmp cpe wait timeout seconds</td>
<td>By default, the close-wait timer is 30 seconds.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enter CWMP view.</td>
<td>cwmp</td>
<td>N/A</td>
</tr>
<tr>
<td>3. Enable NAT traversal.</td>
<td>cwmp cpe stun enable</td>
<td>By default, NAT traversal is disabled on the CPE.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enter CWMP view.</td>
<td>cwmp</td>
<td>N/A</td>
</tr>
<tr>
<td>3. Specify an SSL client policy.</td>
<td>ssl client-policy policy-name</td>
<td>By default, no SSL client policy is specified.</td>
</tr>
</tbody>
</table>

Displaying and maintaining CWMP

Execute display commands in any view.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display CWMP configuration.</td>
<td>display cwmp configuration</td>
</tr>
<tr>
<td>Display the current status of CWMP.</td>
<td>display cwmp status</td>
</tr>
</tbody>
</table>

CWMP configuration example

Network requirements

As shown in Figure 3, use H3C 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>
<thead>
<tr>
<th>Item</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preferred ACS URL</td>
<td><a href="http://10.185.10.41:8080/acs">http://10.185.10.41:8080/acs</a></td>
</tr>
<tr>
<td>ACS username</td>
<td>Admin</td>
</tr>
<tr>
<td>ACS password</td>
<td>12345</td>
</tr>
</tbody>
</table>

Table 25 shows the ACS attributes for the CPEs to connect to the ACS.

Table 26 lists serial numbers of the CPEs.

<table>
<thead>
<tr>
<th>Room</th>
<th>Device</th>
<th>Serial number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Device A</td>
<td>210231A95YH10C000045</td>
</tr>
<tr>
<td></td>
<td>Device B</td>
<td>210235AOLNH12000010</td>
</tr>
<tr>
<td></td>
<td>Device C</td>
<td>210235AOLNH12000015</td>
</tr>
<tr>
<td>B</td>
<td>Device D</td>
<td>210235AOLNH12000017</td>
</tr>
<tr>
<td></td>
<td>Device E</td>
<td>210235AOLNH12000020</td>
</tr>
<tr>
<td></td>
<td>Device F</td>
<td>210235AOLNH12000022</td>
</tr>
</tbody>
</table>

Configuration procedure

Configuring the ACS

1. Log in to the ACS:
1. Launch a Web browser on the ACS configuration terminal.

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**

      ![Image of CPE authentication user configuration page]

      **Query CPE Authentication Users**
      - Username: [Field]
      - 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**
         - LIF
         - 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**

      ![Image of Adding a CPE user account]

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.
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

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.
After the CPE is added successfully, a success message is displayed, as shown in Figure 9.

5. Configure the system settings of the ACS, as shown in Figure 10.
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**

   - On the Configuration Templates page, click Import….
   - 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.
   - Repeat the previous two steps to configure a configuration template for equipment room B’s device class.
After the configuration template is added successfully, a success message is displayed, as shown in Figure 13.

Figure 13 Configuration templates
e. Select Service > BIMS > Configuration Management > Software Library from the top navigation bar.

Figure 14 Configuring software library

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

7. Add auto-deployment tasks:
   a. Select Service > BIMS > Configuration Management > Deployment Guide from the top navigation bar.
   
c. On the **Auto Deploy Configuration** page, click Select Class.

**Figure 17 Configuring auto deployment**

```
<table>
<thead>
<tr>
<th>Tips</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Configuration Template</td>
</tr>
<tr>
<td>Folder</td>
</tr>
<tr>
<td>File Name</td>
</tr>
</tbody>
</table>

**Set Task Attributes**

- Task Name: Task2013-11-97 19:23:49
- Task Type: Auto Deploy Configuration
- Description: 

**Deployment Strategy**

- File Type to be Deployed: Running Configuration

**Select CPE Class**

- [DEVICE_A] (Selected Class)

No match found.

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Vendor</th>
<th>Class Description</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OK Cancel
```

d. On the **Device Class** page, select **Device_A**, and then click **OK**.
e. On the Auto Deploy Configuration page, click OK.
   A success message is displayed, as shown in Figure 19.

Figure 19 Deployment task

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 H3C 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

205
# Enable DHCP server on VLAN-interface 1.
[DHCP_server] interface vlan-interface 1
[DHCP_server-Vlan-interface1] dhcp select server
[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:
2. On the Device Interaction Log page, verify that the configuration has been deployed on the CPEs.

Figure 20 Verifying the configuration deployment status
Configuring PoE

Overview

Power over Ethernet (PoE) enables a network device to supply power to terminals over twisted pair cables. The device supports IEEE 802.3af and IEEE 802.3at.

As shown in Figure 21, a PoE system includes the following elements:

- **PoE power supply**—The PoE power supply provides power for the entire PoE system.
- **PSE**—A power sourcing equipment (PSE) detects and classifies powered devices (PDs), supplies power to PDs, and monitors the PD power and connection status. PSEs include endpoint PSEs and midspan PSEs.

  H3C PSEs are endpoint PSEs. An H3C PSE can be a device with only one built-in PSE, or it can be a PoE-capable interface card or subcard on a device. A device with multiple PSEs uses PSE IDs to identify different PSEs. The `display poe device` command displays the mapping between a PSE ID and the slot or subslot number of a PSE.

- **PI**—A power interface (PI) is a PoE-capable Ethernet interface on a PSE.
- **PD**—A PD receives power from the PSE.

  PDs include IP telephones, APs, portable chargers, POS terminals, and Web cameras.

  You can also connect a PD to a redundant power source for reliability.

Figure 21 PoE system diagram

PoE configuration task list

You can configure a PI directly on the port or by configuring a PoE profile and applying the PoE profile to the PI. To configure one PI, configure it on the port. To configure multiple PIs in batches, use the PoE profile.

Before configuring PoE, make sure the PoE power supply and PSE are operating correctly. Otherwise, either you cannot configure PoE or the PoE configuration cannot take effect.

To configure PoE, perform the following tasks:
### Enabling PoE for a PI

The system only supplies power to and reserves power for PDs connected to PoE-enabled PIs.

You can enable PoE for a PI if the PI will not result in PSE power overload. PSE overload occurs when the sum of the power consumption of all PIs exceeds the maximum power of the PSE. The maximum PSE power is configurable.

If the PI will result in PSE power overload, you can enable PoE for the PI only when PI power management is enabled. For more information about PI power management, see "Configuring PI power management."

The PSE supplies power over the Category 3 or Category 5 twisted pair cable connected to a PI in the following modes:

- **Over signal wires**—The PSE uses data pairs (pins 1, 2, 3, and 6) to supply DC power to PDs.
- **Over spare wires**—The PSE uses spare pairs (pins 4, 5, 7, and 8) to supply DC power to PDs.

The switch series transmits power over signal wires.

#### NOTE:

A PSE can supply power to a PD directly only when the PSE and PD use the same power transmission mode. If the PSE and PD use different power transmission modes, you must change the order of the lines in the twisted pair cable to supply power to the PD.

To enable PoE for a PI:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter system view.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Enter PI view.</td>
<td>interface interface-type interface-number</td>
<td>N/A</td>
</tr>
<tr>
<td>3. Enable PoE for the PI.</td>
<td>poe enable</td>
<td>By default, PoE is disabled for a PI.</td>
</tr>
<tr>
<td>4. (Optional.) Configure a description for the PD connected to the PI.</td>
<td>poe pd-description text</td>
<td>By default, no description is available for the PD connected to the PI.</td>
</tr>
</tbody>
</table>
Enabling nonstandard PD detection

PDs are classified into nonstandard PDs and standard PDs. Standard PDs are compliant with IEEE 802.3af or IEEE 802.3at. PSEs supply power to nonstandard PDs only after you enable nonstandard PD detection.

To enable nonstandard PD detection:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable nonstandard PD detection.</td>
<td>poe legacy enable pse pse-id</td>
</tr>
</tbody>
</table>

Configuring the PD detection mode

Only the LSP7POEB and LSP7POED PSEs support this feature. To verify the PSE model, see the PSE Model field in the output from the display poe pse command.

The device detects PDs in one of the following modes:

- **None**—PDs that are correctly connected to the device without causing short circuit can be supplied with power. Do not use this detection mode when a PI connects to a non-PD device. If you do so, the device tries to supply power to the non-PD device and the non-PD device might be damaged.
- **Simple**—PDs that comply with basic requirements of 802.af or 802.at can be supplied with power.
- **Strict**—PDs that comply with all requirements of 802.af or 802.at can be supplied with power.

This feature can take effect for standard and nonstandard PDs. For the feature to take effect for nonstandard PDs, enable detection for nonstandard PDs by using the poe legacy enable command before you use this feature.

To configure the PD detection mode:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enter PI view.</td>
<td>interface interface-type interface-number</td>
</tr>
<tr>
<td>3.</td>
<td>Configure the PD detection mode.</td>
<td>poe detection-mode { none</td>
</tr>
</tbody>
</table>

Configuring the maximum PI power

The maximum PI power is the maximum power that a PI can provide to the connected PD. If the PD requires more power than the maximum PI power, the PI does not supply power to the PD.

A PSE uses LLDP to negotiate the PI power with a PD after you perform the following tasks:

- Enable PoE and LLDP on the PSE.
- Enable PoE and LLDP, and specify the dot3-tlv power keyword in the lldp tlv-enable command on the PI.

For more information about LLDP, see Layer 2—LAN Switching Configuration Guide.
To configure the maximum PI power:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>interface interface-type interface-number</td>
<td>N/A</td>
</tr>
<tr>
<td>3.</td>
<td>poe max-power max-power</td>
<td>The default maximum PI power is 30000 milliwatts.</td>
</tr>
</tbody>
</table>

### Configuring PI power management

PI power management enables the PSE to perform priority-based PI power management in PSE power overload situations. In descending order, the power-supply priority levels of a PI are critical, high, and low. The PD priority is determined by the priority of the PI to which the PD is connected. All PSEs use the same PI power management mechanism.

If you do not enable PI power management, the PSE might become overloaded. PSE power overload triggers the PoE self-protection mechanism to stop power supply to all PDs.

If you enable PI power management, the PSE stops power supply to existing PDs causing overload or performs priority-based operations for new PDs causing overload:

<table>
<thead>
<tr>
<th>Priority of the new PD</th>
<th>PSE operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>The PSE does not supply power to a new PD.</td>
</tr>
</tbody>
</table>
| High                  | • If low-priority PDs exist, the PSE stops power supply to the existing low-priority PDs, and supplies power to the new PD.  
• If no low-priority PDs exist, the PSE does not supply power to the new PD. |
| Critical              | • If low-priority or high-priority PDs exist, the PSE stops power supply to the existing low-priority or high-priority PDs, and supplies power to the new PD.  
• If no low-priority or high-priority PDs exist, the PSE does not supply power to the new PD. |

**NOTE:**
Configuration for PIs whose power is preempted remains unchanged.

If multiple new PDs require power supply, the PSE supplies power to PDs in priority descending order. For PDs with the same priority, the one with the smallest PD ID takes precedence.

If multiple existing PDs need to be stopped with power supply, the PSE stops power supply to PDs in priority ascending order. For PDs with the same priority, the one with the greatest ID takes precedence.

The PSE guarantees its critical PIs uninterruptable power by reserving guaranteed PSE power. If you want a PI to be allocated with uninterruptable power, configure the PI with critical priority. Otherwise, configure the PI with high or low priority to ensure that other PIs can be supplied with power.

Before you configure a PI with critical priority, make sure the remaining guaranteed power is greater than or equal to the maximum power of the PI. The remaining guaranteed PSE power is the maximum PSE power minus the maximum power for PoE-enabled and PoE-disabled critical PIs.

To configure PI power management:
Configuring PoE monitoring

When the PoE monitoring function is enabled, the system monitors PSEs and PDs in real time. If a specific value exceeds the threshold, the system automatically takes self-protection measures.

If a PSE starts or stops power supply to a PD, the system automatically sends a notification message. For more information, see "Configuring SNMP".

Configuring PSE power monitoring

The system monitors PSE power utilization and sends notification messages when PSE power utilization exceeds or drops below the threshold. If PSE power utilization crosses the threshold multiple times in succession, the system sends notification messages only for the first crossing. For more information about the notification message, see "Configuring SNMP".

To configure a PSE power alarm threshold:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>system-view</td>
<td>N/A</td>
</tr>
<tr>
<td>2.</td>
<td>poe utilization-threshold utilization-threshold-value pse pse-id</td>
<td>By default, the power alarm threshold for the PSE is 80%.</td>
</tr>
</tbody>
</table>

Configuring a PI by using a PoE profile

A PoE profile is a collection of configurations that contain multiple PoE features.

You can configure a PI either on the port or by using a PoE profile. Follow these guidelines when you configure parameters for a PI:

- The **poe max-power max-power** and **poe priority { critical | high | low }** commands must be configured in the same method.
- If you configure a parameter twice with different methods, only the first configuration takes effect. To validate the second configuration, delete the first configuration.

The PoE profile is preferable for PI configuration in batches and PD-specific PI configuration.

- You can apply a PoE profile to multiple PIs. PIs configured by the same PoE profile have the same PoE features.
- You can define PoE configurations for a PD in a PoE profile, and apply the PoE profile to the PI to which the PD connects. This avoids reconfiguration when the PD is moved to another PI.
Configuring a PoE profile

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Create a PoE profile, and enter PoE profile view.</td>
<td>poe-profile profile-name [ index ]</td>
</tr>
<tr>
<td>3.</td>
<td>Enable PoE.</td>
<td>poe enable</td>
</tr>
<tr>
<td>4.</td>
<td>(Optional.) Configure the maximum PI power.</td>
<td>poe max-power max-power</td>
</tr>
<tr>
<td>5.</td>
<td>(Optional.) Configure PI priority.</td>
<td>poe priority { critical</td>
</tr>
</tbody>
</table>

Applying a PoE profile

You can apply a PoE profile in system view or PI view. If you perform the operation in both views, the most recent operation takes effect. To apply a PoE profile to multiple PIs, perform the operation in system view. If a PoE profile is applied, it cannot be deleted or modified before you remove its application.

A PoE profile can be applied to multiple PIs, but a PI can have only one PoE profile.

Applying a PoE profile to multiple PIs in system view

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
</tr>
<tr>
<td>2.</td>
<td>Apply a PoE profile to one or multiple PIs.</td>
</tr>
</tbody>
</table>

Applying a PoE profile to a PI in PI view

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
</tr>
<tr>
<td>2.</td>
<td>Enter PI view.</td>
</tr>
<tr>
<td>3.</td>
<td>Apply the PoE profile to the interface.</td>
</tr>
</tbody>
</table>

Upgrading PSE firmware in service

You can upgrade the PSE firmware in service in either of the following modes:

- **Refresh mode**—Updates the PSE firmware without deleting it. You can use the refresh mode in most cases.
- **Full mode**—Deletes the current PSE firmware and reloads a new one. Use the full mode if the PSE firmware is damaged and you cannot execute any PoE commands.

If the PSE firmware upgrade fails because of interruption such as a device reboot, you can restart the device and upgrade it in full mode again.

To upgrade the PSE firmware in service:
Step | Command
--- | ---
1. Enter system view. | system-view
2. Upgrade the PSE firmware in service. | poe update { full | refresh } filename [ pse pse-id ]

## Displaying and maintaining PoE

Execute **display** commands in any view.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display PSE information.</td>
<td>display poe device [ slot slot-number ]</td>
</tr>
<tr>
<td>Display the power supplying information for the specified PI.</td>
<td>display poe interface [ interface-type interface-number ]</td>
</tr>
<tr>
<td>Display power information for PIs.</td>
<td>display poe interface power [ interface-type interface-number ]</td>
</tr>
<tr>
<td>Display PSE information.</td>
<td>display poe pse [ pse-id ]</td>
</tr>
<tr>
<td>Display the power supplying information for all PIs on a PSE.</td>
<td>display poe pse pse-id interface</td>
</tr>
<tr>
<td>Display power information for all PIs on a PSE.</td>
<td>display poe pse pse-id interface power</td>
</tr>
<tr>
<td>Display all information about the PoE profile.</td>
<td>display poe-profile [ index index</td>
</tr>
<tr>
<td>Display all information about the PoE profile applied to the specified PI.</td>
<td>display poe-profile interface interface-type interface-number</td>
</tr>
</tbody>
</table>

## PoE configuration example

### Network requirements

As shown in **Figure 22**, configure PoE to meet the following requirements:

- Enable the device to supply power to IP telephones and APs.
- Enable the device to supply power to IP telephones first when overload occurs.
- Allocate AP B a maximum power of 9000 milliwatts.

### Figure 22 Network diagram
Configuration procedure

# Enable PoE on GigabitEthernet 1/0/1, GigabitEthernet 1/0/2, and GigabitEthernet 1/0/3, and configure their power supply priority as critical.

```plaintext
<PSE> system-view
[PSE] interface gigabitethernet 1/0/1
[PSE-GigabitEthernet1/0/1] poe enable
[PSE-GigabitEthernet1/0/1] poe priority critical
[PSE-GigabitEthernet1/0/1] quit
[PSE] interface gigabitethernet 1/0/2
[PSE-GigabitEthernet1/0/2] poe enable
[PSE-GigabitEthernet1/0/2] poe priority critical
[PSE-GigabitEthernet1/0/2] quit
[PSE] interface gigabitethernet 1/0/3
[PSE-GigabitEthernet1/0/3] poe enable
[PSE-GigabitEthernet1/0/3] poe priority critical
[PSE-GigabitEthernet1/0/3] quit

# Enable PoE on GigabitEthernet 1/0/11 and GigabitEthernet 1/0/12, and configure the maximum power of GigabitEthernet 1/0/12 as 9000 milliwatts.

[PSE] interface gigabitethernet 1/0/11
[PSE-GigabitEthernet1/0/11] poe enable
[PSE-GigabitEthernet1/0/11] quit
[PSE] interface gigabitethernet 1/0/12
[PSE-GigabitEthernet1/0/12] poe enable
[PSE-GigabitEthernet1/0/12] poe max-power 9000
```

Verifying the configuration

# Connect the IP telephones and APs to the PSE to verify that they can obtain power and operate correctly. (Details not shown.)

Troubleshooting PoE

Failure to set the priority of a PI to critical

Symptom

Power supply priority configuration for a PI failed.

Analysis

Possible reasons include:

- The guaranteed power remained for the PI is less than the maximum power required by the PI.
- The priority of the PI is already set.

Solution

To resolve the problem:

- In the first case, either increase the maximum PSE power or reduce the maximum power of the PI if the remaining guaranteed power of the PSE cannot be modified.
- In the second case, remove the priority that is already configured.
Failure to apply a PoE profile to a PI

Symptom
PoE profile application for a PI failed.

Analysis
Possible reasons include:
• Some configurations in the PoE profile are already configured.
• Some configurations in the PoE profile do not meet the configuration requirements of the PI.
• Another PoE profile is already applied to the PI.

Solution
To resolve the problem:
• In the first case, remove the original configurations.
• In the second case, modify the configurations in the PoE profile.
• In the third case, remove the application of the undesired PoE profile from the PI.
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

<table>
<thead>
<tr>
<th>NETCONF layer</th>
<th>XML layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>Configuration data, status data, and statistics information</td>
<td>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.</td>
</tr>
<tr>
<td>Operations</td>
<td>&lt;get&gt;, &lt;get-config&gt;, &lt;edit-config&gt;…</td>
<td>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.”</td>
</tr>
<tr>
<td>RPC</td>
<td>&lt;rpc&gt;, &lt;rpc-reply&gt;</td>
<td>The RPC layer provides a simple, transport-independent framing mechanism for encoding RPCs. The &lt;rpc&gt; and &lt;rpc-reply&gt; elements are used to enclose NETCONF requests and responses (data at the operations layer and the content layer).</td>
</tr>
</tbody>
</table>
| 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 in the perform NETCONF-over-SOAP operations. In FIPS mode, you can log in through SSH, or the console port to perform NETCONF operations at the CLI. You can also log in through HTTPS to perform NETCONF operations in the perform NETCONF-over-SOAP operations. |
NETCONF message format

NETCONF

IMPORTANT:
When configuring NETCONF in XML view, you must add 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 the end mark 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
<?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.h3c.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:

```xml
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <auth:Authentication env:mustUnderstand="1" xmlns:auth="http://www.h3c.com/netconf/base:1.0">
      <auth:AuthInfo>800207F0120020C</auth:AuthInfo>
    </auth:Authentication>
  </env:Header>
</env:Envelope>
```
<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.h3c.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

<table>
<thead>
<tr>
<th>Configuration tool</th>
<th>Login method</th>
<th>Remarks</th>
</tr>
</thead>
</table>
| 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 interface   | N/A           | To use this method, you must enable NETCONF over SOAP. By default, the device cannot interpret custom interfaces' URLs. For the device to interpret these URLs, you must encode the NETCONF messages sent from a custom interface 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

<table>
<thead>
<tr>
<th>Task at a glance</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional.) Enabling NETCONF over SOAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Enabling NETCONF over SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Enabling NETCONF logging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Required.) Establishing a NETCONF session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Subscribing to event notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Locking/unlocking the configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Performing the get/get-bulk operation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Performing the get-config/get-bulk-config operation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Performing the edit-config operation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Saving, rolling back, and loading the configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Filtering data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Performing CLI operations through NETCONF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Retrieving NETCONF session information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Terminating another NETCONF session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional.) Returning to the CLI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enabling NETCONF over SOAP

NETCONF over SOAP encapsulates NETCONF messages into SOAP messages and transmits the SOAP messages over HTTP or HTTPS. You can use a custom user interface to establish a NETCONF over SOAP session to the device and perform NETCONF operations.

To enable NETCONF over SOAP:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable NETCONF over SOAP.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enable NETCONF over SOAP over HTTP (not available in FIPS mode):</td>
<td>netconf soap http enable</td>
</tr>
<tr>
<td></td>
<td>• Enable NETCONF over SOAP over HTTPS:</td>
<td>netconf soap https enable</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable NETCONF over SSH.</td>
<td>netconf ssh server enable</td>
</tr>
<tr>
<td>3.</td>
<td>Specify a port to listen for NETCONF over SSH connections.</td>
<td>netconf ssh server port number</td>
</tr>
</tbody>
</table>

Enabling NETCONF logging

NETCONF logging generates logs for different NETCONF operation sources and NETCONF operations.

To enable NETCONF logging:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enter system view.</td>
<td>system-view</td>
</tr>
<tr>
<td>2.</td>
<td>Enable NETCONF logging.</td>
<td>netconf log source { all</td>
</tr>
</tbody>
</table>

Establishing a NETCONF session

After a NETCONF session is established, the device automatically sends its capabilities to the client. You must send the capabilities of the client to the device before you can perform any other NETCONF operations.

The device supports a maximum of 32 NETCONF sessions. If the upper limit is reached, new NETCONF users cannot access the device.

Before performing a NETCONF operation, make sure no other users are configuring or managing the device. If multiple users simultaneously configure or manage the device, the result might be different from what you expect.

Entering XML view

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter XML view.</td>
<td>xml</td>
<td>Available in user view.</td>
</tr>
</tbody>
</table>
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
```

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:

```xml
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
        <capability>
            capability-set
        </capability>
    </capabilities>
</hello>
```

Use a pair of `<capability>` and `</capability>` tags to enclose each capability set.

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
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" 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.h3c.com/netconf/event:1.0">
                <Code>code</Code>
                <Group>group</Group>
                <Severity>severity</Severity>
            </event>
        </filter>
    </create-subscription>
</rpc>
```
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>` parameter 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
<rpc-reply message-id="100" 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
<rpc-reply message-id="100" 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**

1. Enter XML view.

```xml
<Sysname> xml
```

2. Notify the device of the NETCONF capabilities supported on the client.

```xml
<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="100" 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="100">
  <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.h3c.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.h3c.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
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" 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
<?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>
```

**Unlocking the configuration**

# Copy the following text to the client to unlock the configuration:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" 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
<?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>
```

**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.
# Notify the device of the NETCONF capabilities supported on the client.

```xml
<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
<rpc message-id="100" 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
<rpc-reply message-id="100" 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
<rpc-reply message-id="100" 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, lock is already held.</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
<?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.h3c.com/netconf/data:1.0">
        Specify the module, submodule, table name, and column name
      </top>
    </filter>
  </getoperation>
</rpc>
```

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
<?xml version="1.0" encoding="UTF-8"?>
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0" xmlns:base="http://www.h3c.com/netconf/base:1.0">
        <Syslog>
          <Logs xc:count="5">
            <Log>
              <Index>10</Index>
            </Log>
          </Logs>
        </Syslog>
      </top>
    </filter>
  </get-bulk>
</rpc>
```
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
<?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 settings, 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
<?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.h3c.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
<?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, test-option, and incremental. For more information about these attributes, see "Appendix A Supported NETCONF operations."

# Copy the following text to perform the <edit-config> operation:
```xml
<?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.h3c.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
<?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.
```xml
<Sysname> xml
```

# Notify the device of the NETCONF capabilities supported on the client.
```xml
<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:

```xml
  <data>
    <top xmlns="http://www.h3c.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>
        </Interfaces>
        <Syslog>
          <LogBuffer>
            <BufferSize>120</BufferSize>
          </LogBuffer>
        </Syslog>
      </Ifmgr>
      <System>
        <Device>
          ...
        </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

# Notify the device of the NETCONF capabilities supported on the client.
<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.h3c.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="100">
  <data>
    <top xmlns="http://www.h3c.com/netconf/config:1.0">
      <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
# Notify the device of the NETCONF capabilities supported on the client.
<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.
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0" xmlns:web="http://www.h3c.com/netconf/base:1.0">
        <Ifmgr>
          <Interfaces web:count="1">
            <Interface>
              <IfIndex>3</IfIndex>
              <Name>GigabitEthernet1/0/2</Name>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>

Verifying the configuration
If the client receives the following text, the get-bulk operation is successful:
  <data>
    <Ifmgr>
      <Interfaces>
        <Interface>
          <IfIndex>3</IfIndex>
        </Interface>
      </Interfaces>
    </Ifmgr>
  </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

# Notify the device of the NETCONF capabilities supported on the client.
<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.
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <top xmlns="http://www.h3c.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
<?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.

Performing the save, rollback, or load operation consumes a lot of system resources. Do not perform these operations when the system resources are heavily occupied.

Saving the configuration

# Copy the following text to the client to save the device configuration to a specified file:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="100" 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 .cfg extension. If the file name is not specified, the configuration is saved to the main next-startup configuration file by default.

After receiving the save request, the device returns a response in the following format if the save operation is successful:

```xml
<?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>
```

Rolling back the configuration based on a configuration file

# Copy the following text to the client to roll back the configuration:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <rollback>
      <file>Specify the configuration file name</file>
   </rollback>
</rpc>
```

After receiving the rollback request, the device returns a response in the following format if the rollback operation is successful:

```xml
<?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>
```
Rolling back the configuration based on a rollback point

You can roll back the running configuration based on a rollback point when one of the following situations occurs:

- A NETCONF client sends a rollback request.
- The NETCONF session idle time is longer than the rollback idle timeout time.
- A NETCONF client is unexpectedly disconnected from the device.

To roll back the configuration based on a rollback point, perform the following tasks:

1. Lock the system.
   
   Multiple users might simultaneously use NETCONF to configure the device. As a best practice, lock the system before rolling back the configuration to prevent other users from modifying the running configuration.

2. Mark the beginning of a rollback operation. For more information, see "Performing the save-point/begin operation."

3. Edit the device configuration. For more information, see "Performing the edit-config operation."

4. Configure the rollback point. For more information, see "Performing the save-point/commit operation."

   You can repeat this step to configure multiple rollback points.

5. Roll back the configuration based on the rollback point. For more information, see "Performing the save-point/rollback operation."

   The configuration can also be automatically rolled back based on the most recently configured rollback point when the NETCONF session idle time is longer than the rollback idle timeout time.

6. End the rollback configuration. For more information, see "Performing the save-point/end operation."

7. Release the lock.

Performing the save-point/begin operation

# Copy the following text to the client to mark the beginning of a rollback operation based on a rollback point:

```xml
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <begin>
      <confirm-timeout>100</confirm-timeout>
    </begin>
  </save-point>
</rpc>
```

The <confirm-timeout> parameter specifies the rollback idle timeout time in the range of 1 to 65535 seconds (the default is 600 seconds). This parameter is optional.

After receiving the begin request, the device returns a response in the following format if the begin operation is successful:

```xml
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <save-point>
      <commit>
        <commit-id>1</commit-id>
      </commit>
    </save-point>
  </data>
</rpc-reply>
```
Performing the save-point/commit operation

The system supports a maximum of 50 rollback points. When the limit is reached, you must specify the **force** attribute to overwrite the earliest rollback point.

# Copy the following text to the client to configure the rollback point:
<pre><code>&lt;rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 &lt;save-point>
   &lt;commit>
     &lt;label>SUPPORT VLAN&lt;/label>
     &lt;comment>VLANs 1 to 100 and interfaces.&lt;/comment>
   &lt;/commit&gt;
 &lt;/save-point&gt;
&lt;/rpc&gt;
</code></pre>

The **<label>** and **<comment>** parameters are optional.

After receiving the commit request, the device returns a response in the following format if the commit operation is successful:
<pre><code>&lt;rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 &lt;data>
   &lt;save-point>
     &lt;commit&gt;
       &lt;commit-id>2&lt;/commit-id&gt;
     &lt;/commit&gt;
   &lt;/save-point&gt;
 &lt;/data&gt;
&lt;/rpc-reply&gt;
</code></pre>

Performing the save-point/rollback operation

# Copy the following text to the client to roll back the configuration:
<pre><code>&lt;rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 &lt;save-point&gt;
   &lt;rollback&gt;
     &lt;commit-id/&gt;
     &lt;commit-index/&gt;
     &lt;commit-label/&gt;
   &lt;/rollback&gt;
 &lt;/save-point&gt;
&lt;/rpc&gt;
</code></pre>

The **<commit-id>** parameter uniquely identifies a rollback point.

The **<commit-index>** parameter specifies 50 most recently configured rollback points. The value of 0 indicates the most recently configured one and 49 indicates the earliest configured one.

The **<commit-label>** parameter exclusively specifies a label for a rollback point. The label is not required for a rollback point.

Specify one of these parameters to roll back the specified configuration. If no parameter is specified, this operation rolls back configuration based on the most recently configured rollback point.

After receiving the rollback request, the device returns a response in the following format if the rollback operation is successful:
<pre><code>&lt;rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 &lt;data>
   &lt;save-point&gt;
     &lt;rollback&gt;
     &lt;/rollback&gt;
   &lt;/save-point&gt;
 &lt;/data&gt;
&lt;/rpc-reply&gt;
</code></pre>
Performing the save-point/end operation

# Copy the following text to the client to end the rollback configuration:
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <end/>
  </save-point>
</rpc>

After receiving the end request, the device returns a response in the following format if the end operation is successful:
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>

Performing the save-point/get-commits operation

# Copy the following text to the client to get the rollback point configuration records:
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <get-commits>
      <commit-id/>
      <commit-index/>
      <commit-label/>
    </get-commits>
  </save-point>
</rpc>

Specify one of the <commit-id>, <commit-index>, and <commit-label> parameters to get the specified rollback point configuration records. If no parameter is specified, this operation gets records for all rollback point settings. The following text is a <save-point>/<get-commits> request example:
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <get-commits>
      <commit-label>SUPPORT VLAN</commit-label>
    </get-commits>
  </save-point>
</rpc>

After receiving the get commits request, the device returns a response in the following format if the get commits operation is successful:
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <save-point>
      <commit-information>
        <CommitID>2</CommitID>
        <TimeStamp>Thu Oct 30 11:30:28 1980</TimeStamp>
        <UserName>test</UserName>
        <Label>SUPPORT VLAN</Label>
      </commit-information>
    </save-point>
  </data>
</rpc-reply>
Performing the save-point/get-commit-information operation

# Copy the following text to the client to get the system configuration data corresponding to a rollback point:

```xml
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <get-commit-information>
      <commit-information>
        <commit-id/>
        <commit-index/>
        <commit-label/>
      </commit-information>
      <compare-information>
        <commit-id/>
        <commit-index/>
        <commit-label/>
      </compare-information>
    </get-commit-information>
  </save-point>
</rpc>
```

Specify one of the `<commit-id>`, `<commit-index>`, and `<commit-label>` parameters to get the configuration data corresponding to the specified rollback point. The `<compare-information>` parameter is optional. If no parameter is specified, this operation gets the configuration data corresponding to the most recently configured rollback point. The following text is a `<save-point>`/<get-commit-information> request example:

```xml
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <get-commit-information>
      <commit-information>
        <commit-label>SUPPORT VLAN</commit-label>
      </commit-information>
    </get-commit-information>
  </save-point>
</rpc>
```

After receiving the get-commit-information request, the device returns a response in the following format if the get-commit-information operation is successful:

```xml
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <save-point>
      <commit-information>
        <content>
          ...
          interface vlan 1
          ...
        </content>
      </commit-information>
    </save-point>
  </data>
</rpc-reply>
```
Loading the configuration

After you perform the load operation, the loaded settings are merged into the current configuration:

- New settings are directly loaded.
- Settings 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
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" 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 `.cfg` extension. If the file name is not specified, the configuration is saved to the main next-startup configuration file by default.

After receiving the load request, the device returns a response in the following format if the load operation is successful:

```xml
<?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>
```

Example for saving the configuration

Network requirements
Save the current configuration to configuration file `my_config.cfg`.

Configuration procedure

# Enter XML view.
```bash
<Sysname> xml
```

# Notify the device of the NETCONF capabilities supported on the client.
```bash
<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 configuration file `my_config.cfg`.
```bash
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" 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
<?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>
```

Filtering data

You can define a filter to filter information when you perform a get, get-bulk, get-config, or get-bulk-config operation. Data filtering includes the following types:

- **Table-based filtering**—Filters table information.
- **Column-based filtering**—Filters information for a single column.

For table-based filtering to take effect, you must configure table-based filtering before column-based filtering.

Table-based filtering

You can specify a match criterion for the row attribute `filter` to implement table-based filtering, for example, IP address filtering. The namespace is `http://www.h3c.com/netconf/base:1.0`. For information about the support for table-based match, see NETCONF XML API documents.

# Copy the following text to the client to retrieve the longest data with IP address 1.1.1.0 and mask length 24 from the IPv4 routing table:

```xml
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Route>
          <Ipv4Routes>
            <RouteEntry h3c:filter="IP 1.1.1.0 MaskLen 24 longer"/>
          </Ipv4Routes>
        </Route>
      </top>
    </filter>
  </get>
</rpc>
```

Column-based filtering

Column-based filtering includes full match filtering, regular expression match filtering, and conditional match filtering. Full match filtering has the highest priority and conditional match filtering has the lowest priority. When more than one filtering criterion is specified, the one with the highest priority takes effect.

Full match filtering

You can specify an element value in an XML message to implement full match filtering. 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="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get>
      <filter type="subtree">
         <top xmlns="http://www.h3c.com/netconf/data:1.0">
            <Ifmgr>
               <Interfaces>
                  <Interface>
                     <AdminStatus>1</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 filtering. The system returns only configuration data that matches this attribute name. The XML message equivalent to the above element-value-based full match filtering is as follows:

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
   <get-config>
      <source>
         <running/>
      </source>
      <filter type="subtree">
         <top xmlns="http://www.h3c.com/netconf/config:1.0">
            <Ifmgr>
               <Interfaces>
                  <Interface data:AdminStatus="1"/>
               </Interfaces>
            </Ifmgr>
         </top>
      </filter>
   </get-config>
</rpc>

The above examples show that both element-value-based full match filtering and attribute-name-based full match filtering can retrieve the same index and column information for all interfaces in up state.

Regular expression match filtering

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="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
   <get-config>
      <source>
         <running/>
      </source>
      <filter type="subtree">
         <top xmlns="http://www.h3c.com/netconf/config:1.0">
            <Ifmgr>
               <Interfaces>
                  <Interface regExp="[^A-Z]"/>
               </Interfaces>
            </Ifmgr>
         </top>
      </filter>
   </get-config>
</rpc>
Conditional match filtering

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Operator</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>More than</td>
<td><code>match=&quot;more:value&quot;</code></td>
<td>More than the specified value. The supported data types include date, digit, and character string.</td>
</tr>
<tr>
<td>Less than</td>
<td><code>match=&quot;less:value&quot;</code></td>
<td>Less than the specified value. The supported data types include date, digit, and character string.</td>
</tr>
<tr>
<td>Not less than</td>
<td><code>match=&quot;notLess:value&quot;</code></td>
<td>Not less than the specified value. The supported data types include date, digit, and character string.</td>
</tr>
<tr>
<td>Not more than</td>
<td><code>match=&quot;notMore:value&quot;</code></td>
<td>Not more than the specified value. The supported data types include date, digit, and character string.</td>
</tr>
<tr>
<td>Equal</td>
<td><code>match=&quot;equal:value&quot;</code></td>
<td>Equal to the specified value. The supported data types include date, digit, character string, OID, and BOOL.</td>
</tr>
<tr>
<td>Not equal</td>
<td><code>match=&quot;notEqual:value&quot;</code></td>
<td>Not equal to the specified value. The supported data types include date, digit, character string, OID, and BOOL.</td>
</tr>
<tr>
<td>Include</td>
<td><code>match=&quot;include:string&quot;</code></td>
<td>Includes the specified string. The supported data types include only character string.</td>
</tr>
<tr>
<td>Not include</td>
<td><code>match=&quot;exclude:string&quot;</code></td>
<td>Excludes the specified string. The supported data types include only character string.</td>
</tr>
<tr>
<td>Start with</td>
<td><code>match=&quot;startswith:string&quot;</code></td>
<td>Starts with the specified string. The supported data types include character string and OID.</td>
</tr>
<tr>
<td>End with</td>
<td><code>match=&quot;endwith:string&quot;</code></td>
<td>Ends with the specified string. The supported data types include only character string.</td>
</tr>
</tbody>
</table>

# Copy the following text to the client to retrieve extension information about the entity of which the CPU usage is more than 50%:

```xml
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Device>
          <ExtPhysicalEntities>
            <Entity>
```
Example for filtering data with regular expression match

Network requirements
Retrieve all data including Gigabit in the Description column of the Interfaces table under the Ifmgr module.

Configuration procedure
# Enter XML view.
<Sysname> xml
# Notify the device of the NETCONF capabilities supported on the client.
<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 Gigabit in the Description column of the Interfaces table under the Ifmgr module.
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <Description reg:regExp="Gigabit"/>
            </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"?>
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

# Notify the device of the NETCONF capabilities supported on the client.
<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:h3c="http://www.h3c.com/netconf/base:1.0">
  <data>
    <top xmlns="http://www.h3c.com/netconf/data:1.0">
      <Ifmgr>
        <Interfaces>
          <Interface>
            <IfIndex>2681</IfIndex>
            <Description>GigabitEthernet1/0/1 Interface</Description>
          </Interface>
          <Interface>
            <IfIndex>2682</IfIndex>
            <Description>GigabitEthernet1/0/2 Interface</Description>
          </Interface>
          <Interface>
            <IfIndex>2683</IfIndex>
            <Description>GigabitEthernet1/0/3 Interface</Description>
          </Interface>
          <Interface>
            <IfIndex>2684</IfIndex>
            <Description>GigabitEthernet1/0/4 Interface</Description>
          </Interface>
        </Interfaces>
      </Ifmgr>
    </top>
  </data>
</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:h3c="http://www.h3c.com/netconf/base:1.0" message-id="100">
  <data>
    <top xmlns="http://www.h3c.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="100" 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
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" 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 vlan` command to the device.

#### Configuration procedure

1. # Enter XML view.
   ```xml
   <Sysname> xml
   </Sysname>
   
   # Notify the device of the NETCONF capabilities supported on the client.
   <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 vlan` command:
   ```xml
   <?xml version="1.0" encoding="UTF-8"?>
   <rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <CLI>
       <Execution>display vlan</Execution>
     </CLI>
   </rpc>
   ```

#### Verifying the configuration

If the client receives the following text, the operation is successful:

```xml
<sysname>display vlan
Total VLANs: 1
The VLANs include:
```

245
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
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" 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
<?xml version="1.0" encoding="UTF-8" ?>
<rpc-reply message-id="100" 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
```

# Notify the device of the NETCONF capabilities supported on the client.
```
<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
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" 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:
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.

Configuration procedure

# Copy the following message to the client to terminate the specified NETCONF session:

```
<rpc message-id="100" 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="100" 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
```

# Notify the device of the NETCONF capabilities supported on the client.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
```

247
<capability>
    urn:ietf:params:netconf:base:1.0
</capability>
</capabilities>
<hello>
    # Terminate the session with session ID 2.
    <rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <kill-session>
            <session-id>2</session-id>
        </kill-session>
    </rpc>
</hello>

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="100" 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="100" 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="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
Configuring the packet capture

To use the packet capture feature, you must install the feature image by using the `boot-loader` command. For more information about the command, see *Fundamentals Command Reference*.

Overview

The packet capture feature captures incoming packets that are to be forwarded in CPU. The feature displays the captured packets on the terminal in real time, and allows you to save the captured packets to a .pcap file for future analysis. Packet capture can read both .pcap and .pcapng files.

Filter elements

Packet capture supports capture filters and display filters. You can use expressions to match packets to capture or display.

A capture or display filter contains a keyword string or multiple keyword strings that are connected by operators.

Keywords include the following types:

- **Qualifiers**—Fixed keyword strings. For example, you must use the `ip` qualifier to specify the IPv4 protocol.
- **Variables**—Values supplied by users in the required format. For example, you can set an IP address to 2.2.2.2 or any other valid values.

A variable must be modified by one or multiple qualifiers. For example, to capture any packets sent from the host at 2.2.2.2, use the filter `src host 2.2.2.2`.

Operators include the following types:

- **Logical operators**—Perform logical operations, such as the AND operation.
- **Arithmetic operators**—Perform arithmetic operations, such as the ADD operation.
- **Relational operators**—Indicate the relation between keyword strings. For example, the `=` operator indicates equality.

This document provides basic information about these elements. For more information about capture and display filters, go to the following websites:

- [http://wiki.wireshark.org/CaptureFilters](http://wiki.wireshark.org/CaptureFilters).

Capture filter keywords

*Table 30* and *Table 31* describe the qualifiers and variables for capture filters, respectively.

**Table 30 Qualifiers for capture filters**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Matches a protocol. If you do not specify a protocol qualifier, the filter matches any supported protocols.</td>
<td><code>arp</code>—Matches ARP. <code>icmp</code>—Matches ICMP. <code>ip</code>—Matches IPv4. <code>ip6</code>—Matches IPv6. <code>tcp</code>—Matches TCP. <code>udp</code>—Matches UDP.</td>
</tr>
<tr>
<td>Category</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Direction   | Matches packets based on its source or destination location (an IP address or port number). If you do not specify a direction qualifier, the src or dst qualifier applies. | - src—Matches the source IP address field.  
- dst—Matches the destination IP address field.  
- src or dst—Matches the source or destination IP address field.  
NOTE: The src or dst qualifier applies if you do not specify a direction qualifier. For example, port 23 is equivalent to src or dst port 23. |
| Type        | Specifies the direction type.                                               | - host—Matches the IP address of a host.  
- net—Matches an IP subnet.  
- port—Matches a service port number.  
- portrange—Matches a service port range.  
NOTE: The host qualifier applies if you do not specify any type qualifier. For example, src 2.2.2.2 is equivalent to src host 2.2.2.2. To specify an IPv6 subnet, you must specify the net qualifier. |
| Others      | Any other qualifiers than the previously described qualifiers.             | - broadcast—Matches broadcast packets.  
- multicast—Matches multicast and broadcast packets.  
- less—Matches packets that are less than or equal to a specific size.  
- greater—Matches packets that are greater than or equal to a specific size.  
- len—Matches the packet length.  
- vlan—Matches VLAN packets. |

NOTE: The broadcast, multicast, and all protocol qualifiers cannot modify variables.

Table 31 Variable types for capture filters

<table>
<thead>
<tr>
<th>Variable type</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>Represented in binary, octal, decimal, or hexadecimal notation.</td>
<td>The port 23 expression matches traffic sent to or from port number 23.</td>
</tr>
<tr>
<td>Integer range</td>
<td>Represented by hyphenated integers.</td>
<td>The portrange 100-200 expression matches traffic sent to or from any ports in the range of 100 to 200.</td>
</tr>
<tr>
<td>IPv4 address</td>
<td>Represented in dotted decimal notation.</td>
<td>The src 1.1.1.1 expression matches traffic sent from the IPv4 host at 1.1.1.1.</td>
</tr>
<tr>
<td>IPv6 address</td>
<td>Represented in colon hexadecimal notation.</td>
<td>The dst host 1::1 expression matches traffic sent to the IPv6 host at 1::1.</td>
</tr>
<tr>
<td>IPv4 subnet</td>
<td>Represented by an IPv4 network ID or an IPv4 address with a mask.</td>
<td>Both of the following expressions match traffic sent to or from the IPv4 subnet 1.1.1.0/24:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- src 1.1.1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- src net 1.1.1.0/24.</td>
</tr>
<tr>
<td>Variable type</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IPv6 network segment</td>
<td>Represented by an IPv6 address with a prefix length.</td>
<td>The dst net 1::/64 expression matches traffic sent to the IPv6 network 1::/64.</td>
</tr>
</tbody>
</table>

**Capture filter operators**

Capture filters support logical operators (Table 32), arithmetic operators (Table 33), and relational operators (Table 34). Logical operators can use both alphanumeric and nonalphanumeric symbols. The arithmetic and relational operators can use only nonalphanumeric symbols.

Logical operators are left associative. They group from left to right. The not operator has the highest priority. The and and or operators have the same priority.

**Table 32 Logical operators for capture filters**

<table>
<thead>
<tr>
<th>Nonalphanumeric symbol</th>
<th>Alphanumeric symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>not</td>
<td>Reverses the result of a condition. Use this operator to capture traffic that matches the opposite value of a condition. For example, to capture non-HTTP traffic, use not port 80.</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>and</td>
<td>Joins two conditions. Use this operator to capture traffic that matches both conditions. For example, to capture non-HTTP traffic that is sent to or from 1.1.1.1, use host 1.1.1.1 and not port 80.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 33 Arithmetic operators for capture filters**

<table>
<thead>
<tr>
<th>Nonalphanumeric symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Adds two values.</td>
</tr>
<tr>
<td>-</td>
<td>Subtracts one value from another.</td>
</tr>
<tr>
<td>*</td>
<td>Multiplies one value by another.</td>
</tr>
<tr>
<td>/</td>
<td>Divides one value by another.</td>
</tr>
<tr>
<td>&amp;</td>
<td>Returns the result of the bitwise AND operation on two integral values in binary form.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;&lt;</td>
<td>Performs the bitwise left shift operation on the operand to the left of the operator. The right-hand operand specifies the number of bits to shift.</td>
</tr>
<tr>
<td>&gt;&gt;</td>
<td>Performs the bitwise right shift operation on the operand to the left of the operator. The right-hand operand specifies the number of bits to shift.</td>
</tr>
<tr>
<td>Nonalphanumeric symbol</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>[ ]</td>
<td>Specifies a byte offset relative to a protocol layer. This offset indicates the byte where the matching begins. You must enclose the offset value in the brackets and specify a protocol qualifier. For example, ip[6] matches the seventh byte of payload in IPv4 packets (the byte that is six bytes away from the beginning of the IPv4 payload).</td>
</tr>
</tbody>
</table>

**Table 34 Relational operators for capture filters**

<table>
<thead>
<tr>
<th>Nonalphanumeric symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal to. For example, ip[6]=0x1c matches an IPv4 packet if its seventh byte of payload is equal to 0x1c.</td>
</tr>
<tr>
<td>!=</td>
<td>Not equal to. For example, len!=60 matches a packet if its length is not equal to 60 bytes.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than. For example, len&gt;100 matches a packet if its length is greater than 100 bytes.</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than. For example, len&lt;100 matches a packet if its length is less than 100 bytes.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to. For example, len&gt;=100 matches a packet if its length is greater than or equal to 100 bytes.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to. For example, len&lt;=100 matches a packet if its length is less than or equal to 100 bytes.</td>
</tr>
</tbody>
</table>

**Display filter keywords**

Table 35 and Table 36 describe the qualifiers and variables for display filters, respectively.

**Table 35 Qualifiers for display filters**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Matches a protocol.</td>
<td>• eth—Matches Ethernet. • ftp—Matches FTP. • http—Matches HTTP. • icmp—Matches ICMP. • ip—Matches IPv4. • ipv6—Matches IPv6. • tcp—Matches TCP. • telnet—Matches Telnet. • udp—Matches UDP.</td>
</tr>
<tr>
<td>Packet field</td>
<td>Matches a field in packets by using a dotted string in the protocol.field[level1-subfield]…[leveln-subfield] format.</td>
<td>• tcp.flags.syn—Matches the SYN bit in the flags field of TCP. • tcp.port—Matches the source or destination port field.</td>
</tr>
</tbody>
</table>

**NOTE:**
The protocol qualifiers cannot modify variables.
### Table 36 Variable types for display filters

<table>
<thead>
<tr>
<th>Variable type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>Represented in binary, octal, decimal, or hexadecimal notation.</td>
</tr>
<tr>
<td></td>
<td>For example, to display IP packets that are less than or equal to 1500 bytes, use one of the following expressions:</td>
</tr>
<tr>
<td></td>
<td>• ip.len le 1500.</td>
</tr>
<tr>
<td></td>
<td>• ip.len le 02734.</td>
</tr>
<tr>
<td></td>
<td>• ip.len le 0x436.</td>
</tr>
<tr>
<td>Boolean</td>
<td>This variable type has two values: true or false.</td>
</tr>
<tr>
<td></td>
<td>This variable type applies if you use a packet field string alone to identify the presence of a field in a packet.</td>
</tr>
<tr>
<td></td>
<td>• If the field is present, the match result is true. The filter displays the packet.</td>
</tr>
<tr>
<td></td>
<td>• If the field is not present, the match result is false. The filter does not display the packet.</td>
</tr>
<tr>
<td></td>
<td>For example, to display TCP packets that contain the SYN field, use tcp.flags.syn.</td>
</tr>
<tr>
<td>MAC address (six bytes)</td>
<td>Uses colons (:), dots (.), or hyphens (-) to break up the MAC address into two or four segments.</td>
</tr>
<tr>
<td></td>
<td>For example, to display packets that contain a destination MAC address of ffff.ffff.ffff, use one of the following expressions:</td>
</tr>
<tr>
<td></td>
<td>• eth.dst==ff:ff:ff:ff:ff:ff.</td>
</tr>
<tr>
<td></td>
<td>• eth.dst==ff-ff-ff-ff-ff-ff.</td>
</tr>
<tr>
<td></td>
<td>• eth.dst ==ffff.ffff.ffff.</td>
</tr>
<tr>
<td>IPv4 address</td>
<td>Represented in dotted decimal notation.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>• To display IPv4 packets that are sent to or from 192.168.0.1, use ip.addr==192.168.0.1.</td>
</tr>
<tr>
<td></td>
<td>• To display IPv4 packets that are sent to or from 129.111.0.0/16, use ip.addr==129.111.0.0/16.</td>
</tr>
<tr>
<td>IPv6 address</td>
<td>Represented in colon hexadecimal notation.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>• To display IPv6 packets that are sent to or from 1::1, use ipv6.addr==1::1.</td>
</tr>
<tr>
<td></td>
<td>• To display IPv6 packets that are sent to or from 1::/64, use ipv6.addr==1::/64.</td>
</tr>
<tr>
<td>String</td>
<td>Character string.</td>
</tr>
<tr>
<td></td>
<td>For example, to display HTTP packets that contain the string HTTP/1.1 for the request version field, use http.request version==&quot;HTTP/1.1&quot;.</td>
</tr>
</tbody>
</table>

### Display filter operators

Display filters support logical operators (Table 37) and relational operators (Table 38). Both operator types can use alphanumeric and nonalphanumeric symbols.

Logical operators are left associative. They group from left to right. Table 37 displays logical operators by priority, from the highest to the lowest. The and and or operators have the same priority.
# Logical operators for display filters

<table>
<thead>
<tr>
<th>Nonalphanumeric symbol</th>
<th>Alphanumeric symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ]</td>
<td>No alphanumeric symbol is available.</td>
<td>Used with protocol qualifiers. For more information, see &quot;The proto[…] expression.&quot;</td>
</tr>
<tr>
<td>!</td>
<td>not</td>
<td>Displays packets that do not match the condition connected to this operator.</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>and</td>
<td>Joins two conditions. Use this operator to display traffic that matches both conditions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Relational operators for display filters

<table>
<thead>
<tr>
<th>Nonalphanumeric symbol</th>
<th>Alphanumeric symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>eq</td>
<td>Equal to. For example, <code>ip.src==10.0.0.5</code> displays packets with the source IP address as 10.0.0.5.</td>
</tr>
<tr>
<td>!=</td>
<td>ne</td>
<td>Not equal to. For example, <code>ip.src!=10.0.0.5</code> displays packets whose source IP address is not 10.0.0.5.</td>
</tr>
<tr>
<td>&gt;</td>
<td>gt</td>
<td>Greater than. For example, <code>frame.len&gt;100</code> displays frames with a length greater than 100 bytes.</td>
</tr>
<tr>
<td>&lt;</td>
<td>lt</td>
<td>Less than. For example, <code>frame.len&lt;100</code> displays frames with a length less than 100 bytes.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>ge</td>
<td>Greater than or equal to. For example, <code>frame.len ge 0x100</code> displays frames with a length greater than or equal to 256 bytes.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>le</td>
<td>Less than or equal to. For example, <code>frame.len le 0x100</code> displays frames with a length less than or equal to 256 bytes.</td>
</tr>
</tbody>
</table>

---

# Building a capture filter

This section provides the most commonly used expression types for capture filters.

## Logical expression

Use this type of expression to capture packets that match the result of logical operations.

Logical expressions contain keywords and logical operators. For example:

- **not port 23 and not port 22**—Captures packets with a port number that is not 23 or 22.
- **port 23 or icmp**—Captures packets with a port number 23 or ICMP packets.
In a logical expression, a qualifier can modify more than one variable connected by its nearest logical operator. For example, to capture packets sourced from IPv4 address 192.168.56.1 or IPv4 network 192.168.27, use either of the following expressions:

- `src 192.168.56.1 or 192.168.27`
- `src 192.168.56.1 or src 192.168.27`

The `expr relop expr` expression

Use this type of expression to capture packets that match the result of arithmetic operations. This expression contains keywords, arithmetic operators (`expr`), and relational operators (`relop`). For example, `len+100>=200` captures packets that are greater than or equal to 100 bytes.

The `proto [ expr size ]` expression

Use this type of expression to capture packets that match the result of arithmetic operations on a number of bytes relative to a protocol layer. This type of expression contains the following elements:

- `proto`—Specifies a protocol layer.
- `[]`—Performs arithmetic operations on a number of bytes relative to the protocol layer.
- `expr`—Specifies the arithmetic expression.
- `size`—Specifies the byte offset. This offset indicates the number of bytes relative to the protocol layer. The operation is performed on the specified bytes. The offset is set to 1 byte if you do not specify an offset.

For example, `ip[0]&0xf !=5` captures an IP packet if the result of ANDing the first byte with 0x0f is not 5.

To match a field, you can specify a field name for `expr:size`. For example, `icmp[icmptype]=0x08` captures ICMP packets that contain a value of 0x08 in the Type field.

The `vlan vlan_id` expression

Use this type of expression to capture 802.1Q tagged VLAN traffic. This type of expression contains the `vlan vlan_id` keywords and logical operators. The `vlan_id` variable is an integer that specifies a VLAN ID. For example, `vlan 1 and ip6` captures IPv6 packets in VLAN 1.

To capture 802.1Q tagged traffic, you must use the `vlan vlan_id` expression prior to any other expressions. An expression matches untagged packets if it does not follow a `vlan vlan_id` expression. For example:

- `vlan 1 and !tcp`—Captures VLAN 1-tagged non-TCP packets.
- `icmp and vlan 1`—Captures untagged ICMP packets that are VLAN 1 tagged. This expression does not capture any packets because no packets can be both tagged and untagged.

Building a display filter

This section provides the most commonly used expression types for display filters.

**Logical expression**

Use this type of expression to display packets that match the result of logical operations. Logical expressions contain keywords and logical operators. For example, `ftp or icmp` displays all FTP packets and ICMP packets.

**Relational expression**

Use this type of expression to display packets that match the result of comparison operations.
Relational expressions contain keywords and relational operators. For example, `ip.len<=28` displays IP packets that contain a value of 28 or fewer bytes in the length field.

**Packet field expression**

Use this type of expression to display packets that contain a specific field.

Packet field expressions contain only packet field strings. For example, `tcp.flags.syn` displays all TCP packets that contain the SYN bit field.

**The `proto[... ]` expression**

Use this type of expression to display packets that contain specific field values.

This type of expression contains the following elements:

- `proto`—Specifies a protocol layer or packet field.
- `[... ]`—Matches a number of bytes relative to a protocol layer or packet field. Values for the bytes to be matched must be a hexadecimal integer string. The expression in brackets can use the following formats:
  - `[n:m]`—Matches a total of `m` bytes after an offset of `n` bytes from the beginning of the specified protocol layer or field. To match only 1 byte, you can use both `[n]` and `[n:1]` formats. For example, `eth.src[0:3]==00:00:83` matches an Ethernet frame if the first three bytes of its source MAC address are 0x00, 0x00, and 0x83. The `eth.src[2] == 83` expression matches an Ethernet frame if the third byte of its source MAC address is 0x83.
  - `[n-m]`—Matches a total of `(m-n+1)` bytes, starting from the `(n+1)`th byte relative to the beginning of the specified protocol layer or packet field. For example, `eth.src[1-2]==00:83` matches an Ethernet frame if the second and third bytes of its source MAC address are 0x00 and 0x83, respectively.

### Configuring packet capture

#### Saving captured packets to a file

Perform this task to capture incoming packets on an interface and save the captured packets to a file. To display the captured packets, use the `packet-capture read` command. To stop the capture while it is capturing packets, press **Ctrl+C**. There might be a delay for the capture to stop because of heavy traffic.

To save the captured packets to a file:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save the captured packets to a file.</td>
<td>`packet-capture interface interface-type interface-number [capture-filter capt-expression</td>
<td>limit-captured-frames limit</td>
</tr>
</tbody>
</table>

### Filtering packet data to display

To stop the capture while it is capturing packets, press **Ctrl+C**. There might be a delay for the capture to stop because of heavy traffic.
To filter packet data to display:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter packet data to display.</td>
<td>packet-capture interface interface-type interface-number [ capture-filter capt-expression</td>
<td>After this command is executed, you cannot configure other commands from the CLI until the capture completes capturing packets or it is stopped.</td>
</tr>
<tr>
<td></td>
<td>capture-filter limit</td>
<td>display-filter limit-captured-frames limit</td>
</tr>
</tbody>
</table>

Displaying the contents in a packet file

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display the contents in a packet file.</td>
<td>packet-capture read filepath [ display-filter disp-expression ] [ raw</td>
<td>{ brief</td>
</tr>
</tbody>
</table>

Packet capture configuration examples

Filtering packet data to display configuration example

**Network requirements**

On Switch A, capture the following incoming IP packets on GigabitEthernet 1/0/1:

- Packets forwarded through the CPU.
- Packets that are sourced from 192.168.56.1 0 and forwarded through chips.

**Configuration procedure**

```
# Create an IPv4 advanced ACL to match packets that are sourced from 192.168.56.1 0.
<SwitchA> system-view
[SwitchA] acl number 3000
[SwitchA-acl-adv-3000] rule permit ip source 192.168.56.1 0
[SwitchA-acl-adv-3000] quit

# Configure a traffic behavior to mirror traffic to the CPU.
[SwitchA] traffic behavior behavior1
[SwitchA-behavior-behavior1] mirror-to cpu
[SwitchA-behavior-behavior1] quit

# Configure a traffic class to use the ACL to match traffic.
[SwitchA] traffic classifier classifier1
[SwitchA-classifier-classifier1] if-match acl 3000
[SwitchA-classifier-classifier1] quit

# Associate the traffic class with the traffic behavior in a QoS policy.
[SwitchA] qos policy user1
[SwitchA-qospolicy-user1] classifier classifier1 behavior behavior1
[SwitchA-qospolicy-user1] quit

# Apply the QoS policy to the incoming traffic of GigabitEthernet 1/0/1.
[SwitchA] interface gigabitethernet 1/0/1
```

257
[SwitchA-GigabitEthernet1/0/1] qos apply policy user1 inbound
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] quit

# Capture incoming traffic on GigabitEthernet 1/0/1.
<SwitchA> packet-capture interface gigabitethernet1/0/1
Capturing on 'GigabitEthernet1/0/1'

1   0.000000 192.168.56.1 -> 192.168.56.2 TCP 62 6325 > telnet [SYN] Seq=0 Win
   =65535 Len=0 MSS=1460 SACK_PERM=1
2   0.000061 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=1 Ack
   =1 Win=65535 Len=0
3   0.024370 192.168.56.1 -> 192.168.56.2 TELNET 60 Telnet Data ...
4   0.024449 192.168.56.1 -> 192.168.56.2 TELNET 78 Telnet Data ...
5   0.025766 192.168.56.1 -> 192.168.56.2 TELNET 65 Telnet Data ...
6   0.035096 192.168.56.1 -> 192.168.56.2 TELNET 60 Telnet Data ...
7   0.047317 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ack
k=434 Win=65102 Len=0
8   0.050994 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ack
k=436 Win=65100 Len=0
9   0.052401 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ack
k=438 Win=65098 Len=0
10  0.057736 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ack
k=440 Win=65096 Len=0
10 packets captured

Saving captured packets to a file configuration example

Network requirements
On Device A, capture 10 incoming packets on GigabitEthernet 1/0/1, save the packets to a packet file, and display contents in the file.

Configuration procedure

# Capture packets on GigabitEthernet 1/0/1. Set the maximum number of captured packets to 10. Save the packets to the file flash:/a.pcap.
<DeviceA> packet-capture interface gigabitethernet1/0/1 limit-captured-frames 10 write flash:/a.pcap
Capturing on 'GigabitEthernet1/0/1'

1   0.000000 192.168.56.1 -> 192.168.56.2 TCP 62 6325 > telnet [SYN] Seq=0 Win
   =65535 Len=0 MSS=1460 SACK_PERM=1
2   0.000061 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=1 Ack
   =1 Win=65535 Len=0
3   0.024370 192.168.56.1 -> 192.168.56.2 TELNET 60 Telnet Data ...
4   0.024449 192.168.56.1 -> 192.168.56.2 TELNET 78 Telnet Data ...
5   0.025766 192.168.56.1 -> 192.168.56.2 TELNET 65 Telnet Data ...
6   0.035096 192.168.56.1 -> 192.168.56.2 TELNET 60 Telnet Data ...
7   0.047317 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ack
k=434 Win=65102 Len=0
8   0.050994 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ack
k=436 Win=65100 Len=0
9   0.052401 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ack
k=438 Win=65098 Len=0
10  0.057736 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ack
k=440 Win=65096 Len=0
10 packets captured
8  0.050994 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ac
k=436 Win=65100 Len=0
9  0.052401 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ac
k=438 Win=65098 Len=0
10 0.057736 192.168.56.1 -> 192.168.56.2 TCP 60 6325 > telnet [ACK] Seq=42 Ac
k=440 Win=65096 Len=0
Appendix

Appendix A Supported NETCONF operations

Table 39 lists the NETCONF operations available with Comware 7.

Table 39 NETCONF operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>XML example</th>
</tr>
</thead>
</table>
| get       | Retrieves device configuration and state information. | To retrieve device configuration and state information for the Syslog module:  
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Syslog>
          <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:  
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface/>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-config>
</rpc> |
<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>XML example</th>
</tr>
</thead>
<tbody>
<tr>
<td>get-bulk</td>
<td>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.</td>
<td>To retrieve device configuration and state information for all interface:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;rpc message-id=&quot;100&quot; xmlns=&quot;urn:ietf:params:xml:ns:netconf:base:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;get-bulk&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;filter type=&quot;subtree&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;top xmlns=&quot;http://www.h3c.com/netconf/data:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Interfaces xc:count=&quot;5&quot; xmlns:xc=&quot;http://www.h3c.com/netconf/base:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Interface/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Interfaces&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/top&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/filter&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/get-bulk&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/rpc&gt;</td>
</tr>
<tr>
<td>get-bulk-config</td>
<td>Retrieves a number of non-default configuration data entries starting from the data entry next to the one with the specified index.</td>
<td>To retrieve non-default configuration for all interfaces:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;rpc message-id=&quot;100&quot; xmlns=&quot;urn:ietf:params:xml:ns:netconf:base:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;get-bulk-config&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;source&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;running/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/source&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;filter type=&quot;subtree&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;top xmlns=&quot;http://www.h3c.com/netconf/config:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/top&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/filter&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/get-bulk-config&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/rpc&gt;</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
<td>XML example</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| edit-config: incremental | Adds configuration data to a column without affecting the original data. The incremental attribute applies to a list column such as the vlan permitlist column. You can use the incremental attribute for edit-config operations except for the replace operation. Support for the incremental attribute varies by module. For more information, see NETCONF XML API documents. | To add VLANs 1 through 10 to an untagged VLAN list that has untagged VLANs 12 through 15:  
```xml
<rpc message-id="100"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
 <edit-config>
 <target>
  <running/>
 </target>
 <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <top xmlns="http://www.h3c.com/netconf/config:1.0">
   <VLAN xc:operation="merge">
    <HybridInterfaces>
     <Interface>
      <IfIndex>262</IfIndex>
      <UntaggedVlanList h3c: incremental="true">1-10</UntaggedVlanList>
     </Interface>
    </HybridInterfaces>
   </VLAN>
  </top>
 </config>
</edit-config>
</rpc>``` |
<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>XML example</th>
</tr>
</thead>
</table>
| 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:  
  <edit-config>  
    <target>  
      <running/>  
    </target>  
    <config>  
      <top xmlns="http://www.h3c.com/netconf/config:1.0"><Syslog xmlns="http://www.h3c.com/netconf/config:1.0" xc:operation="merge">  
        <LogBuffer>  
          <BufferSize>120</BufferSize>  
        </LogBuffer>  
      </Syslog>  
    </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**. |
| 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**. |
<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>XML example</th>
</tr>
</thead>
<tbody>
<tr>
<td>edit-config: remove</td>
<td>Removes the specified configuration.</td>
<td>The syntax is the same as the edit-config message with the <strong>merge</strong> attribute. Change the operation attribute from <strong>merge</strong> to <strong>remove</strong>.</td>
</tr>
<tr>
<td></td>
<td>• If the specified target has only the table index, the operation removes all configuration of the specified target, and the target itself.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If the specified target has the table index and configuration data, the operation removes the specified configuration data of this target.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If the specified target does not exist, or the XML message does not specify any target, a success message is returned.</td>
<td></td>
</tr>
<tr>
<td>edit-config: delete</td>
<td>Deletes the specified configuration.</td>
<td>The syntax is the same as the edit-config message with the <strong>merge</strong> attribute. Change the operation attribute from <strong>merge</strong> to <strong>delete</strong>.</td>
</tr>
<tr>
<td></td>
<td>• If the specified target has only the table index, the operation removes all configuration of the specified target, and the target itself.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If the specified target has the table index and configuration data, the operation removes the specified configuration data of this target.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If the specified target does not exist, an error message is returned, showing that the target does not exist.</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
<td>XML example</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>edit-config:</td>
<td>Modifies the current configuration of the device using the default operation method.</td>
<td>To issue an empty operation for schema verification purposes:</td>
</tr>
<tr>
<td>default-operation</td>
<td>If you do not specify an operation attribute for an edit-config message, NETCONF uses one of the</td>
<td>&lt;rpc message-id=&quot;100&quot; xmlns=&quot;urn:ietf:params:xml:ns:netconf:base:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>following default operation attributes: merge, create, delete, and replace. Your setting of</td>
<td>&lt;edit-config&gt;</td>
</tr>
<tr>
<td></td>
<td>the value for the &lt;default-operation&gt; element takes effect only once. If you do not specify</td>
<td>&lt;target&gt;</td>
</tr>
<tr>
<td></td>
<td>an operation attribute and the default operation method for an &lt;edit-config&gt; message, merge</td>
<td>&lt;running/&gt;</td>
</tr>
<tr>
<td></td>
<td>is always applied.</td>
<td>&lt;/target&gt;</td>
</tr>
<tr>
<td></td>
<td>• merge—The default value for the &lt;default-operation&gt; element.</td>
<td>&lt;default-operation&gt;none&lt;/default-operation&gt;</td>
</tr>
<tr>
<td></td>
<td>• replace—Value used when the operation attribute is not specified and the default operation</td>
<td>&lt;config xmlns:xc=&quot;urn:ietf:params:xml:ns:netconf:base:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>method is specified as replace.</td>
<td>&lt;top xmlns=&quot;http://www.h3c.com/netconf/config:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>• none—Value used when the operation attribute is not specified and the default operation</td>
<td>&lt;Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td>method is specified as none. If this value is specified, the edit-config operation is used</td>
<td>&lt;Interfaces&gt;</td>
</tr>
<tr>
<td></td>
<td>only for schema verification rather than issuing a configuration. If the schema verification</td>
<td>&lt;Interface&gt;</td>
</tr>
<tr>
<td></td>
<td>is passed, a successful message is returned. Otherwise, an error message is returned.</td>
<td>&lt;Index&gt;262&lt;/Index&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Description&gt;222222&lt;/Description&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Interfaces&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/top&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/config&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/edit-config&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/rpc&gt;</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
<td>XML example</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>edit-config:</td>
<td>Determines the action to take in case of a configuration error.</td>
<td>To issue the configuration for two interfaces with the error-option element value as continue-on-error:</td>
</tr>
</tbody>
</table>
| error-option       | The error-option element has one of the following values:                   | <rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
|                    | • **stop-on-error**—Stops the operation on error and returns an error message. |   <edit-config>
|                    | • **continue-on-error**—Continues the operation on error and returns an error message. |     <target>
|                    | • **rollback-on-error**—Rolls back the configuration.                        |       <error-option>continue-on-error</error-option>
|                    |                                                                             |       <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
|                    |                                                                             |         <top xmlns="http://www.h3c.com/netconf/config:1.0">
|                    |                                                                             |           <Ifmgr xc:operation="merge">
|                    |                                                                             |             <Interfaces>
|                    |                                                                             |               <Interface>
|                    |                                                                             |                 <Index>262</Index>
|                    |                                                                             |                 <Description>222</Description>
|                    |                                                                             |                 <ConfigSpeed>1024</ConfigSpeed>
|                    |                                                                             |                 <ConfigDuplex>1</ConfigDuplex>
|                    |                                                                             |               </Interface>
|                    |                                                                             |             </Interfaces>
|                    |                                                                             |           </Ifmgr>
|                    |                                                                             |         </top>
|                    |                                                                             |     </config>
|                    |                                                                             |   </edit-config>
|                    |                                                                             | </rpc>
<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>XML example</th>
</tr>
</thead>
<tbody>
<tr>
<td>edit-config:</td>
<td>Determines whether to issue a configuration item in the edit-configure operation. The test-option element has one of the following values:</td>
<td>To issue the configuration for an interface for test purposes:</td>
</tr>
<tr>
<td>test-option</td>
<td>• <strong>test-then-set</strong>—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.</td>
<td>&lt;rpc message-id=&quot;100&quot; xmlns=&quot;urn:ietf:params:xml:ns:netconf:base:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>• <strong>set</strong>—Directly performs the set operation without the validation test.</td>
<td>&lt;edit-config&gt;</td>
</tr>
<tr>
<td></td>
<td>• <strong>test-only</strong>—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.</td>
<td>&lt;target&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;running/&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/target&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;test-option&gt;test-only&lt;/test-option&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;config xmlns:xc=&quot;urn:ietf:params:xml:ns:netconf:base:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;top xmlns=&quot;http://www.h3c.com/netconf/config:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Ifmgr xc:operation=&quot;merge&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Interfaces&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Interface&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Index&gt;262&lt;/Index&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Interface&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Interfaces&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/top&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/config&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/edit-config&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/rpc&gt;</td>
</tr>
<tr>
<td>action</td>
<td>Issues actions that are not for configuring data, for example, reset action.</td>
<td>To clear statistics information for all interfaces:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;rpc message-id=&quot;100&quot; xmlns=&quot;urn:ietf:params:xml:ns:netconf:base:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;action&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;top xmlns=&quot;http://www.h3c.com/netconf/action:1.0&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;ClearAllIfStatistics&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Clear&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Clear&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/ClearAllIfStatistics&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/Ifmgr&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/top&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/action&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/rpc&gt;</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
<td>XML example</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| lock      | Locks the configuration data that can be changed by the edit-config operation. Other settings are not limited by the lock operation. This lock operation locks only the settings made through NETCONF sessions, rather than those through other protocols, for example, SNMP. | To lock the configuration:  
```xml
<rpc message-id="100" 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:  
```xml
<rpc message-id="100" 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:  
```xml
<rpc message-id="100" 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:  
```xml
<rpc message-id="100" 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:  
```xml
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <kill-session>
    <session-id>1</session-id>
  </kill-session>
</rpc>
``` |
<table>
<thead>
<tr>
<th><strong>Operation</strong></th>
<th><strong>Description</strong></th>
<th><strong>XML example</strong></th>
</tr>
</thead>
</table>
| **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:  
```xml  
<rpc message-id="100" 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 you do not specify a file, the running configuration is saved to the main next-startup configuration file. | To save the running configuration to file **test.cfg**:  
```xml  
<rpc message-id="100" 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 file **a1.cfg** to the current configuration of the device:  
```xml  
<rpc message-id="100" 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 file **1A.cfg**:  
```xml  
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">  
  <rollback>  
    <file>1A.cfg</file>  
  </rollback>  
</rpc>  
``` |
Index

A
access control
- NMM NTP access control rights configuration, 15
- NMM NTP peer ACL, 10
- NMM NTP query ACL, 10
- NMM NTP security, 10
- NMM NTP server ACL, 10
- NMM NTP synchronization ACL, 10
- NMM SNMP MIB, 65
- NMM SNMP MIB role-based, 66
- NMM SNMP MIB view-based, 66

ACL
- NMM NTP peer ACL, 10
- NMM NTP query ACL, 10
- NMM NTP server ACL, 10
- NMM NTP synchronization ACL, 10

ACS
- attribute type, 191
- attribute type (default)(CLI), 193
- attribute type (preferred), 191
- authentication parameters, 193
- autoconnect parameters, 194
- connection interface, 194
- HTTPS SSL client policy, 196
- provision code, 193

address
- NMM ping address reachability determination, 1

agent
- sFlow agent+collector information configuration, 168

alarm
- NMM RMON alarm function, 84
- NMM RMON alarm function configuration, 88
- NMM RMON alarm group sample type, 83
- NMM RMON configuration, 81
- NMM RMON Ethernet statistics group configuration, 85
- NMM RMON group, 82
- NMM RMON history group configuration, 86
- NMM RMON private group, 82
- RMON configuration, 85

applying
- NMM PoE profile, 212
- NMM traffic mirroring QoS policy, 163
- NMM traffic mirroring QoS policy (control plane), 164
- NMM traffic mirroring QoS policy (global), 164
- NMM traffic mirroring QoS policy (interface), 163
- NMM traffic mirroring QoS policy (VLAN), 164

architecture
- NMM NTP, 8

arithmetic
- capture NTP operator, 251

arithmetic operator
- capture filter, 249

assigning
- CWMP ACS attribute (preferred)(CLI), 192
- CWMP ACS attribute (preferred)(DHCP server), 191
- NMM port mirroring monitor port to remote probe VLAN, 155

associating
- NMM IPv6 NTP client/server association mode, 28
- NMM IPv6 NTP multicast association mode, 36
- NMM IPv6 NTP symmetric active/passive association mode, 30
- NMM NTP association mode, 12
- NMM NTP broadcast association mode, 9, 14, 31
- NMM NTP broadcast association mode with authentication, 40
- NMM NTP client/server association mode, 9, 12, 27
- NMM NTP client/server association mode with authentication, 39
- NMM NTP multicast association mode, 9, 14, 33
- NMM NTP symmetric active/passive association mode, 9, 13, 29

authenticating
- CWMP CPE ACS authentication parameters, 193
- NMM NTP, 11
- NMM NTP broadcast authentication, 20
- NMM NTP broadcast mode with authentication, 40
- NMM NTP client/server mode authentication, 16
- NMM NTP client/server mode with authentication, 39
- NMM NTP configuration, 16
- NMM NTP multicast authentication, 22
- NMM NTP security, 10
- NMM NTP symmetric active/passive mode authentication, 18
- NMM SNTP authentication, 45
auto
   CWMP ACS autoconnect parameters, 194
B
bidirectional
   NMM port mirroring, 147
broadcast
   NMM NTP association mode, 31
   NMM NTP broadcast association mode, 9, 14, 20
   NMM NTP broadcast association mode with authentication, 40
   NMM NTP broadcast client configuration, 14
   NMM NTP broadcast mode max number dynamic associations, 25
   NMM NTP broadcast server configuration, 14
buffer
   NMM information center log buffer, 55
C
capturing
   packet configuration, 249
changing
   NMM NETCONF parameter value, 232
classifying
   NMM port mirroring classification, 148
CLI
   NMM EAA configuration, 176, 183
   NMM EAA monitor policy configuration, 180, 183
   NMM EAA monitor policy configuration (CLI-defined with environment variables), 184
   NMM NETCONF CLI operations, 244, 245
   NMM NETCONF return to CLI, 248
client
   NMM NQA client DHCP operation, 94
   NMM NQA client DLSw operation, 104
   NMM NQA client DNS operation, 95
   NMM NQA client FTP operation, 96
   NMM NQA client FTP template, 117
   NMM NQA client history record save, 111
   NMM NQA client HTTP operation, 97
   NMM NQA client HTTP template, 116
   NMM NQA client ICMP echo operation, 93
   NMM NQA client ICMP template, 112, 113
   NMM NQA client operation scheduling, 112
   NMM NQA client SNMP operation, 99
   NMM NQA client statistics collection, 111
   NMM NQA client TCP template, 114
   NMM NQA client template, 112
   NMM NQA client template optional parameters, 118
   NMM NQA client threshold monitoring, 91, 108
   NMM NQA client UDP echo operation, 100
   NMM NQA client UDP jitter operation, 98
   NMM NQA client voice operation, 102
   NMM NQA client+Track collaboration function, 107
   NMM NQA collaboration configuration, 139
   NMM NQA DHCP operation configuration, 121
   NMM NQA DLSw operation configuration, 136
   NMM NQA DNS operation configuration, 122
   NMM NQA DNS template configuration, 142
   NMM NQA enable, 93
   NMM NQA FTP operation configuration, 124
   NMM NQA FTP template configuration, 145
   NMM NQA HTTP operation configuration, 125
   NMM NQA HTTP template configuration, 144
   NMM NQA ICMP echo operation configuration, 120
   NMM NQA operation, 93
   NMM NQA SNMP operation configuration, 128
   NMM NQA TCP operation configuration, 130
   NMM NQA TCP template configuration, 143
   NMM NQA UDP echo operation configuration, 131
   NMM NQA UDP jitter operation configuration, 126
   NMM NQA voice operation configuration, 134
   NMM NTP broadcast client configuration, 14
   NMM NTP multicast client configuration, 14
   NMM SNTP configuration, 44, 46
   NQA client template (UDP), 115
   NQA UDP tracert operation configuration, 132
close-wait timer (CWMP ACS), 195
collaborating
   NMM NQA client+Track collaboration function, 107
   NMM NQA+Track collaboration, 91
collecting
   sFlow agent+collector information configuration, 168
troubleshooting sFlow remote collector cannot receive packets, 171
commom logs
   NMM information center common logs, 48
comparison
capture filter operator, 251
display filter expression, 255
display filter operator, 253
comparison operator
capture filter, 249
conditional match
NETCONF data filtering (column-based), 241
conditional match NETCONF data filtering, 243
configuring
CWMP, 187, 190, 196
CWMP ACS attribute, 191
CWMP ACS attribute (default)(CLI), 193
CWMP ACS attribute (preferred), 191
CWMP ACS autoconnect parameters, 194
CWMP ACS close-wait timer, 195
CWMP ACS connection retry max number, 195
CWMP ACS periodic Inform feature, 194
CWMP CPE ACS authentication parameters, 193
CWMP CPE ACS connection interface, 194
CWMP CPE ACS provision code, 193
CWMP CPE attribute, 193
CWMP CPE NAT traversal, 195
EAA monitor policy ( Tcl), 181
information center log suppression, 60
local port mirroring with multiple monitor ports, 160
max PI power, 209
NMM EAA, 176, 183
NMM EAA environment variable (user-defined), 179
NMM EAA monitor policy, 179
NMM EAA monitor policy (CLI), 180, 183
NMM EAA monitor policy (CLI-defined with environment variables), 184
NMM EAA monitor policy (Tcl), 185
NMM information center, 48, 53, 61
NMM information center trace log file max size, 58
NMM IPv6 NTP client/server association mode, 28
NMM IPv6 NTP multicast association mode, 36
NMM IPv6 NTP symmetric active/passive association mode, 30
NMM Layer 2 remote port mirroring, 153, 158
NMM local port mirroring, 149, 157
NMM local port mirroring group monitor port, 151
NMM local port mirroring group monitor port (in interface view), 151
NMM local port mirroring group monitor port (in system view), 151
NMM local port mirroring group source ports, 150
NMM local port mirroring group source ports (in interface view), 150
NMM local port mirroring group source ports (in system view), 150
NMM NETCONF, 216, 219
NMM NQA, 90, 92, 120
NMM NQA client DHCP operation, 94
NMM NQA client DLSw operation, 104
NMM NQA client DNS operation, 95
NMM NQA client DNS template, 113
NMM NQA client FTP operation, 96
NMM NQA client FTP template, 117
NMM NQA client history record save, 111
NMM NQA client HTTP operation, 97
NMM NQA client HTTP template, 116
NMM NQA client ICMP echo operation, 93
NMM NQA client ICMP template, 112
NMM NQA client operation, 93
NMM NQA client operation optional parameters, 106
NMM NQA client path jitter operation, 105
NMM NQA client SNMP operation, 99
NMM NQA client statistics collection function, 111
NMM NQA client TCP operation, 100
NMM NQA client TCP template, 114
NMM NQA client template, 112
NMM NQA client template optional parameters, 118
NMM NQA client threshold monitoring, 108
NMM NQA client UDP echo operation, 100
NMM NQA client UDP jitter operation, 98
NMM NQA client voice operation, 102
NMM NQA client+Track collaboration function, 107
NMM NQA collaboration, 139
NMM NQA DHCP operation, 121
NMM NQA DLSw operation, 136
NMM NQA DNS operation, 122
NMM NQA DNS template, 142
NMM NQA FTP operation, 124
NMM NQA FTP template, 145
NMM NQA HTTP operation, 125
NMM NQA HTTP template, 144
NMM NQA ICMP echo operation, 120
NMM NQA ICMP template, 141
NMM NQA path ICMP echo operation, 137
NMM NQA server, 92
NMM NQA SNMP operation, 128
NMM NQA TCP operation, 130
NMM NQA TCP template, 143
NMM NQA UDP echo operation, 131
NMM NQA UDP jitter operation, 126
NMM NQA voice operation, 134
NMM NTP, 7, 12
NMM NTP access control rights, 15
NMM NTP association mode, 12
NMM NTP broadcast association mode, 14, 31
NMM NTP broadcast client, 14
NMM NTP broadcast mode authentication, 20
NMM NTP broadcast mode with authentication, 40
NMM NTP broadcast server, 14
NMM NTP client/server association mode, 12, 27
NMM NTP client/server mode authentication, 16
NMM NTP client/server mode with authentication, 39
NMM NTP local clock as reference source, 26
NMM NTP max number dynamic associations, 25
NMM NTP multicast association mode, 14, 33
NMM NTP multicast client, 14
NMM NTP multicast mode authentication, 22
NMM NTP multicast server, 15
NMM NTP optional parameters, 24
NMM NTP packet DSCP value, 26
NMM NTP symmetric active/passive association mode, 13, 29
NMM NTP symmetric active/passive mode authentication, 18
NMM PoE, 213
NMM port mirroring, 157
NMM port mirroring remote destination group monitor port, 154
NMM port mirroring remote destination group remote probe VLAN, 154
NMM remote port mirroring destination group, 153
NMM remote port mirroring source group, 155
NMM remote port mirroring source group egress port, 156
NMM remote port mirroring source group remote probe VLAN, 156
NMM remote port mirroring source group source ports, 155
NMM RMON, 81
NMM RMON alarm function, 84, 88
NMM RMON Ethernet statistics group, 85
NMM RMON history group, 86
NMM RMON statistics function, 83
NMM SNMP basics, 67
NMM SNMP logging, 72
NMM SNMP notification, 72
NMM SNMPv1, 75
NMM SNMPv1 basics, 67
NMM SNMPv2c, 75
NMM SNMPv2c basics, 67
NMM SNMPv3 (RBAC mode), 78
NMM SNMPv3 (VACM mode), 76
NMM SNMPv3 basics, 68
NMM SNTP, 44, 46
NMM SNTP authentication, 45
NMM traffic mirroring, 162, 162, 165
NMM traffic mirroring match criteria, 162
NMM traffic mirroring QoS policy, 163
NMM traffic mirroring traffic behavior, 163
NQA client template (UDP), 115
NQA UDP template configuration, 143
NQA UDP tracer operation, 132
NTP, 27
packet capture, 249, 256, 257
PD detection mode, 209
PL power management, 210
PL using PoE profile, 211
PMM kernel thread deadloop detection, 173
PMM kernel thread starvation detection, 174
PoE, 207, 207
PoE monitoring, 211
PoE profile, 212
PSE power monitoring, 211
RMON, 85
saving captured packets to a file, 258
sFlow, 167, 167, 169
sFlow agent+collector information, 168
sFlow counter sampling, 169
sFlow flow sampling, 168
SNMP, 75
SNMPv1 host notification send, 73
SNMPv2c host notification send, 73
SNMPv3 host notification send, 73
connecting
CWMP ACS connection initiation, 195
CWMP ACS connection retry max number, 195
CWMP CPE ACS connection interface, 194
console
NMM information center log output, 53, 61
constant
capture filter keyword, 249
control plane
NMM traffic mirroring QoS policy application, 164
controlling
NMM NTP access control rights configuration, 15
NMM RMON history control entry, 83
CPE
ACS authentication parameters, 193
ACS connection interface, 194
ACS provision code, 193
attribute type, 193
CWMP ACS autoconnect parameters, 194
NAT traversal, 195
CPE WAN Management Protocol. Use CWMP
CPU
NMM traffic mirroring configuration, 162, 162, 165
creating
NMM local port mirroring local group, 150
NMM port mirroring remote destination group, 153
NMM port mirroring remote source group, 155
NMM RMON Ethernet statistics entry, 83
NMM RMON history control entry, 83
CWMP
ACS, 187
ACS attribute (default)(CLI), 193
ACS attribute (preferred), 191
ACS attribute configuration, 191
ACS autoconnect parameters, 194
ACS HTTPS SSL client policy, 196
basic functions, 187
configuration, 187, 190, 196
CPE, 187
CPE ACS authentication parameters, 193
CPE ACS connection interface, 194
CPE ACS provision code, 193
CPE attribute configuration, 193
CPE NAT traversal, 195
DHCP server, 187
displaying settings, 196
DNS server, 187
enable, 191
how it works, 188
network framework, 187
D
data
NETCONF data entry retrieval (interface table), 231
NMM NETCONF configuration data retrieval (all modules), 228
NMM NETCONF configuration data retrieval (Syslog module), 230
NMM NETCONF filtering (column-based), 239
NMM NETCONF filtering (column-based) (conditional match), 241
NMM NETCONF filtering (column-based) (full match), 239
NMM NETCONF filtering (column-based) (regex match), 240
NMM NETCONF filtering (conditional match), 243
NMM NETCONF filtering (regex match), 242
NMM NETCONF filtering (table-based), 239
deadloop detection (Linux kernel PMM), 173
debugging
NMM feature module, 6
NMM system maintenance, 1
default
NMM information center log default output rules, 49
NMM system information diagnostic log output rules, 49
NMM system information hidden log output rules, 50
NMM system information security log output rules, 49
NMM system information trace log output rules, 50
destination
NMM information center system logs, 49
NMM port mirroring, 147
detecting
PMM kernel thread deadloop detection, 173
PMM kernel thread starvation detection, 174
PoE PD (nonstandard), 209
PoE PD detection mode, 209
determining
NMM address reachability with ping, 1
device
CWMP configuration, 187, 190, 196
local port mirroring configuration with multiple monitor ports, 160
max PI power, 209
NETCONF save-point/begin operation, 234
NETCONF save-point/commit operation, 235
NETCONF save-point/end operation, 236
NETCONF save-point/get-commit-information operation, 237
NETCONF save-point/get-commits operation, 236
NETCONF save-point/rollback operation, 235
NMM information center configuration, 48, 53, 61
NMM information center log output (console), 61, 61
NMM information center log output (Linux log host), 63
NMM information center log output (UNIX log host), 62
NMM information center system log types, 48
NMM IPv6 NTP multicast association mode, 36
NMM Layer 2 remote port mirroring configuration, 153, 158
NMM local port mirroring configuration, 149, 157
NMM local port mirroring group monitor port, 151
NMM local port mirroring group monitor port (in interface view), 151
NMM local port mirroring group monitor port (in system view), 151
NMM NETCONF capability exchange, 221
NMM NETCONF CLI operations, 244, 245
NMM NETCONF configuration, 216, 219
NMM NETCONF configuration lock/unlock, 223, 224
NMM NETCONF edit-config operation, 228
NMM NETCONF get/get-bulk operation, 226
NMM NETCONF get-config/get-bulk-config operation, 227
NMM NETCONF service operations, 225
NMM NETCONF session information retrieval, 246
NMM NETCONF session termination, 247
NMM NQA client operation, 93
NMM NQA server, 92
NMM NTP architecture, 8
NMM NTP broadcast association mode, 31
NMM NTP broadcast mode with authentication, 40
NMM NTP multicast association mode, 33
NMM port mirroring configuration, 147, 157
NMM port mirroring remote destination group, 153
NMM port mirroring remote source group, 155
NMM port mirroring remote source group egress port, 156
NMM port mirroring remote source group remote probe VLAN, 156
NMM port mirroring remote source group source ports, 155
NMM SNMP basics, 67
NMM SNMP FIPS compliance, 67
NMM SNMP MIB, 65
NMM SNMP MIB access control mode, 66
NMM SNMP MIB role-based access control, 66
NMM SNMP MIB view-based access control, 66
NMM SNMP notification, 72
NMM SNMP view-based MIB access control, 65
NMM SNMPv1 basics, 67
NMM SNMPv1 configuration, 75
NMM SNMPv2c basics, 67
NMM SNMPv2c configuration, 75
NMM SNMPv3 basics, 68
NMM SNMPv3 configuration (RBAC mode ), 78
NMM SNMPv3 configuration (VACM mode ), 76
NTP configuration, 27
PI configuration using PoE profile, 211
PoE configuration, 207, 207
SNMP configuration, 75
DHCP
NMM NQA, 90
NMM NQA client DHCP operation, 94
NMM NQA DHCP operation configuration, 121
diagnosing
NMM information center diagnostic log, 48
NMM information center diagnostic log file save, 57
direction
NMM port mirroring (bidirectional), 147
NMM port mirroring (inbound), 147
NMM port mirroring (outbound), 147
disabling
NMM information center interface link up/link down log generation, 60
NMM NTP message receiving, 25
display filter keyword
packet capture, 252
display filter operator
packet capture, 253
displaying
contents in a packet file, 257
CWMP settings, 196
NMM EAA settings, 182
NMM information center, 60
NMM NQA, 119
NMM NTP, 26
NMM PoE, 213
NMM RMON settings, 85
NMM SNMP settings, 74
NMM SNTP, 46
PMM, 172
PMM kernel threads, 174
sFlow, 169
user PMM, 172

DLSw
NMM NQA, 90
NMM NQA client DLSw operation, 104
NMM NQA DLSw operation configuration, 136

DNS
NMM NQA, 90
NMM NQA client DNS operation, 95
NMM NQA client DNS template, 113
NMM NQA DNS operation configuration, 122
NMM NQA DNS template configuration, 142

domain
name system. Use DNS

DSCP
NMM NTP packet value configuration, 26

DSL network
CWMP configuration, 187, 190
duplicate log suppression, 59
dynamic
NMM NTP max number dynamic associations, 25

Dynamic Host Configuration Protocol. Use DHCP

E

EAA
configuration, 176, 183
displaying settings, 182
environment variable configuration (user-defined), 179
event monitor, 176
event monitor policy action, 177
event monitor policy element, 177
event monitor policy environment variable, 178
event monitor policy event types, 177
event monitor policy runtime, 178
event monitor policy user role, 178
event source, 176
how it works, 176
monitor policy, 176
monitor policy configuration, 179
monitor policy configuration (CLI), 180, 183
monitor policy configuration (CLI-defined with environment variables), 184
monitor policy configuration (Tcl), 181, 185
monitor policy suspension, 182
RTM, 176
echo
NMM NQA client ICMP echo operation, 93
NMM NQA client UDP echo operation, 100
NMM NQA ICMP echo operation configuration, 120
NMM NQA UDP echo operation configuration, 131

egress port
NMM Layer 2 remote port mirroring, 147
NMM port mirroring remote source group egress port, 156
electricity
PoE configuration, 207, 207

Embedded Automation Architecture. Use EAA

enabling
CWMP, 191
NETCONF logging, 220
NETCONF over SSH, 220
NMM information center duplicate log suppression, 59
NMM information center synchronous output, 59
NMM NETCONF over SOAP, 219
NMM NQA client, 93
NMM NTP, 12
NMM PoE PI, 208
NMM SNMP notification, 72
NMM SNTP, 44
PoE PD detection (nonstandard), 209

entering
NMM NETCONF XML view, 220

environment
NMM EAA environment variable configuration (user-defined), 179
NMM EAA event monitor policy environment variable, 178
NMM EAA monitor policy configuration (CLI-defined with environment variables), 184

establishing
NMM NETCONF session, 220

Ethernet
CWMP configuration, 187, 190, 196
NMM Layer 2 remote port mirroring configuration, 153
NMM port mirroring configuration, 147, 157
NMM RMON alarm function configuration, 88
NMM RMON Ethernet statistics entry, 83
NMM RMON Ethernet statistics group configuration, 85
NMM RMON history group configuration, 86
NMM RMON statistics group, 81
PoE configuration, 207, 207
power over Ethernet. Use PoE
sFlow configuration, 167, 167, 169

EAA event monitor policy element, 177
NMM EAA configuration, 176, 183
NMM EAA environment variable configuration (user-defined), 179
NMM EAA event monitor, 176
NMM EAA event monitor policy environment variable, 178
NMM EAA event source, 176
NMM EAA monitor policy, 176
NMM NETCONF event notification subscription, 221, 222
NMM RMON event group, 81

exchanging
NMM NETCONF capabilities, 221

F

field
display filter expression, 255
display filter keyword, 252

file
NMM information center diagnostic log output destination, 57
NMM information center log output destination, 55
NMM information center security log file management, 57
NMM information center security log save to file, 56

filtering
column-based filtering, 239
displaying packet data, 256
NMM NETCONF data (conditional match), 243
NMM NETCONF data (regex match), 242
NMM NETCONF data filtering (column-based), 239
NMM NETCONF data filtering (table-based), 239
table-based filtering, 239

FIPS
NMM SNMP FIPS compliance, 67

FIPS compliance
NETCONF, 219
NMM information center, 53

firmware
PoE PSE firmware upgrade, 212

flow
Sampled Flow. Use sFlow format
NMM information center system logs, 50

NMM NETCONF message, 217

FTP
NMM NQA, 90
NMM NQA client FTP operation, 96
NMM NQA client FTP template, 117
NMM NQA FTP operation configuration, 124
NMM NQA FTP template configuration, 145

full match
NETCONF data filtering (column-based), 239

function
NMM RMON alarm function, 84
NMM RMON statistics function, 83

G
generating
NMM information center interface link up/link down log generation, 60

get operation
NMM NETCONF get/get-bulk, 226
NMM NETCONF get-config/get-bulk-config, 227

Get operation (SNMP), 66, 72
group
NMM local port mirroring group monitor port, 151
NMM local port mirroring group monitor port (in interface view), 151
NMM local port mirroring group monitor port (in system view), 151
NMM local port mirroring group source port, 150
NMM local port mirroring group source port (in interface view), 150
NMM local port mirroring group source port (in system view), 150
NMM port mirroring group, 147
NMM RMON, 81
NMM RMON alarm, 82
NMM RMON alarm group sample type, 83
NMM RMON Ethernet statistics, 81
NMM RMON Ethernet statistics group configuration, 85
NMM RMON event, 81
NMM RMON history, 81
NMM RMON history group configuration, 86
NMM RMON private alarm, 82
NMM RMON private alarm group sample type, 83

H

hidden log (information center), 48

history
NMM NQA client history record save, 111
NMM RMON alarm function configuration, 88
NMM RMON group, 81
NMM RMON history control entry, 83
NMM RMON history group configuration, 86, 86

host
NMM information center log output (log host), 54

HTTP
NMM NETCONF over SOAP (HTTP-based), 219
NMM NQA, 90
NMM NQA client HTTP operation, 97
NMM NQA HTTP operation configuration, 125
NMM NQA HTTP template configuration, 144

HTTPS
CWMP ACS HTTPS SSL client policy, 196
NMM NETCONF over SOAP (HTTPS-based), 219

ICMP
NMM NQA, 90
NMM NQA client ICMP echo operation, 93
NMM NQA client ICMP template, 112
NMM NQA collaboration configuration, 139
NMM NQA ICMP echo operation configuration, 120
NMM NQA ICMP template configuration, 141
ping command, 1

identifying
NMM node failure with tracert, 4

implementing
NMM local port mirroring, 148
NMM port mirroring, 148
NMM remote port mirroring, 148

inbound
NMM port mirroring, 147

information center
configuration, 48, 53, 61
diagnostic log default output rules, 49
diagnostic log save to file, 57
displaying, 60
duplicate log suppression, 59
FIPS compliance, 53
hidden log default output rules, 50
interface link up/link down log generation, 60
log default output rules, 49
log minimum storage period, 59
log output (console), 53, 61
log output (Linux log host), 63
log output (log buffer), 55
log output (log host), 54

log output (monitor terminal), 54
log output (UNIX log host), 62
log save to file, 55
maintaining, 60
security log default output rules, 49
security log file management, 57
security log management, 56
security log save to file, 56
synchronous log output, 59
system information log types, 48
system log destinations, 49
system log formats, 50
system log levels, 48
trace log default output rules, 50
trace log file max size, 58

initiating
CWMP ACS connection initiation, 195

Internet
NMM NQA configuration, 90, 92, 120
NMM SNMP basics, 67
NMM SNMP MIB, 65
NMM SNMP MIB access control mode, 66
NMM SNMP notification, 72
NMM SNMPv1 basics, 67
NMM SNMPv1 configuration, 75
NMM SNMPv2c basics, 67
NMM SNMPv2c configuration, 75
NMM SNMPv3 basics, 68
NMM SNMPv3 configuration (RBAC mode ), 78
NMM SNMPv3 configuration (VACM mode ), 76
SNMP configuration, 75

interval
CWMP ACS periodic Inform feature, 194

IP addressing
NMM tracert, 3, 4
NMM tracert node failure identification, 4

IP services
NMM NQA client DHCP operation, 94
NMM NQA client DLSw operation, 104
NMM NQA client DNS operation, 95
NMM NQA client DNS template, 113
NMM NQA client FTP operation, 96
NMM NQA client FTP template, 117
NMM NQA client history record save, 111
NMM NQA client HTTP operation, 97
NMM NQA client HTTP template, 116
NMM NQA client ICMP echo operation, 93
NMM NQA client ICMP template, 112
NMM NQA client operation optional parameters, 106
NMM NQA client operation scheduling, 112
NMM NQA client path jitter operation, 105
NMM NQA client SNMP operation, 99
NMM NQA client statistics collection, 111
NMM NQA client TCP operation, 100
NMM NQA client TCP template, 114
NMM NQA client template optional parameters, 118
NMM NQA client threshold monitoring, 108
NMM NQA client UDP echo operation, 100
NMM NQA client UDP jitter operation, 98
NMM NQA client voice operation, 102
NMM NQA client+Track collaboration function, 107
NMM NQA collaboration configuration, 139
NMM NQA configuration, 90, 92, 120
NMM NQA DHCP operation configuration, 121
NMM NQA DLSw operation configuration, 136
NMM NQA DNS operation configuration, 122
NMM NQA DNS template configuration, 142
NMM NQA FTP operation configuration, 124
NMM NQA FTP template configuration, 145
NMM NQA HTTP operation configuration, 125
NMM NQA HTTP template configuration, 144
NMM NQA ICMP echo operation configuration, 120
NMM NQA ICMP template configuration, 141
NMM NQA path jitter operation configuration, 137
NMM NQA SNMP operation configuration, 128
NMM NQA TCP operation configuration, 130
NMM NQA TCP template configuration, 143
NMM NQA UDP echo operation configuration, 131
NMM NQA UDP jitter operation configuration, 126
NMM NQA voice operation configuration, 134
NQA client template (UDP), 115
NQA UDP template configuration, 143
NQA UDP tracert operation configuration, 132

IPv6
NMM NTP client/server association mode, 28
NMM NTP multicast association mode, 36
NMM NTP symmetric active/passive association mode, 30

K
kernel thread
displaying, 174
Linux process, 172

maintaining, 174
PMM, 173
PMM deadloop detection, 173
PMM starvation detection, 174

keyword
capture filter, 249

L
Layer 2
NMM port mirroring configuration, 147, 157
NMM remote port mirroring configuration, 153, 158

Layer 3
NMM port mirroring configuration, 147, 157
NMM tracert, 3, 4
NMM tracert node failure identification, 4

level
NMM information center system logs, 48

link
NMM information center interface link up/link down log generation, 60

Linux
displaying kernel thread (PMM), 174
kernel thread, 172
kernel thread deadloop detection (PMM), 173
kernel thread PMM, 173
kernel thread starvation detection (PMM), 174
maintaining kernel thread (PMM), 174
NMM information center log host, 63
PMM, 172

loading
NMM NETCONF configuration, 233, 238

local
NMM local port mirroring, 148
NMM local port mirroring configuration, 149
NMM local port mirroring group creation, 150
NMM local port mirroring group monitor port, 151
NMM local port mirroring group monitor port (in interface view), 151
NMM local port mirroring group monitor port (in system view), 151
NMM local port mirroring group source port, 150
NMM local port mirroring group source port (in interface view), 150
NMM local port mirroring group source port (in system view), 150

local clock as reference source, 26
local port mirroring
configuration, 160

locking
NMM NETCONF configuration, 223, 224

logging
information center log minimum storage period, 59
information center log suppression, 60
NETCONF logging enable, 220
NMM information center common logs, 48
NMM information center configuration, 48, 53, 61
NMM information center diagnostic log save to file, 57
NMM information center diagnostic logs, 48
NMM information center duplicate log suppression, 59
NMM information center hidden logs, 48
NMM information center interface link up/link down log generation, 60
NMM information center log default output rules, 49
NMM information center log output (console), 53, 61
NMM information center log output (Linux log host), 63
NMM information center log output (log buffer), 55
NMM information center log output (log host), 54
NMM information center log output (monitor terminal), 54
NMM information center log output (UNIX log host), 62
NMM information center log save to file, 55
NMM information center security log file management, 57
NMM information center security log management, 56
NMM information center security log save to file, 56
NMM information center security logs, 48
NMM information center synchronous log output, 59
NMM information center system log destinations, 49
NMM information center system log formats, 50
NMM information center system log levels, 48
NMM information center trace log file max size, 58
NMM SNMP configuration, 72
NMM system information diagnostic log default output rules, 49
NMM system information hidden log default output rules, 50
NMM system information security log default output rules, 49

NMM system information trace log default output rules, 50

logical
  capture filter expression, 254
capture filter operator, 251
display filter expression, 255
display filter operator, 253

logical operator
  capture filter, 249

M
maintaining
  NMM information center, 60
  NMM PoE, 213
PMM, 172
PMM kernel threads, 174
PMM Linux, 172
process monitoring and maintenance. See PMM Management Information Base. Use MIB managing
  NMM information center security log file, 57
  NMM information center security logs, 56
matching
  NMM NETCONF data filtering (column-based), 239
  NMM NETCONF data filtering (column-based) (conditional match), 241
  NMM NETCONF data filtering (column-based) (full match), 239
  NMM NETCONF data filtering (column-based) (regex match), 240
  NMM NETCONF data filtering (conditional match), 243
  NMM NETCONF data filtering (regex match), 242
  NMM NETCONF data filtering (table-based), 239
  NMM traffic mirroring match criteria, 162
message
  NMM NETCONF format, 217
  NMM NTP message receiving disable, 25
  NMM NTP message source interface, 24
MIB
  NMM role-based access control, 66
  NMM SNMP, 65, 65
  NMM SNMP access control mode, 66
  NMM SNMP Get operation, 66
  NMM SNMP Set operation, 66
  NMM view-based access control, 65, 66
mirroring
  port. See port mirroring
traffic. See traffic mirroring
mode
  NMM NTP broadcast association mode, 9
NMM NTP client/server association mode, 9
NMM NTP multicast association mode, 9
NMM NTP symmetric active/passive association mode, 9
PoE PD detection, 209
PoE PSE firmware upgrade (full mode), 212
PoE PSE firmware upgrade (refresh mode), 212
module
  NMM feature module debug, 6
  NMM information center configuration, 48, 53, 61
  NMM NETCONF configuration data retrieval (all modules), 228
  NMM NETCONF configuration data retrieval (Syslog module), 230
monitor terminal
  NMM information center log output, 54
monitoring
  NMM EAA monitor policy configuration (CLI-defined with environment variables), 184
  NMM NQA threshold monitoring, 91
Monitoring
  PoE monitoring, 211
monitoring
  configuring local mirroring to support multiple monitor ports, 151
  kernel thread PMM, 173
  NMM EAA configuration, 176, 183
  NMM EAA environment variable configuration (user-defined), 179
  NMM EAA event monitor, 176
  NMM EAA event monitor policy environment variable, 178
  NMM EAA monitor policy, 176
  NMM EAA monitor policy configuration, 179
  NMM EAA monitor policy configuration (CLI), 160, 183
  NMM EAA monitor policy configuration (Tcl), 185
  NMM EAA monitor policy suspension, 182
  NMM NQA client threshold monitoring, 108
  PMM Linux, 172
  process monitoring and maintenance.
  See PMM
multicast
  NMM IPv6 NTP multicast association mode, 36
  NMM NTP multicast association mode, 9, 14, 33
  NMM NTP multicast client configuration, 14
  NMM NTP multicast mode authentication, 22
  NMM NTP multicast mode max number dynamic associations, 25
  NMM NTP multicast server configuration, 15
N
NAT
  CWMP CPE NAT traversal, 195
NETCONF
  capability exchange, 221
  CLI operations, 244, 245
  CLI return, 248
  configuration, 216, 219
  configuration data retrieval (all modules), 228
  configuration data retrieval (Syslog module), 230
  configuration load, 233, 238
  configuration lock/unlock, 223, 224
  configuration rollback, 233
  configuration rollback (configuration file-based), 233
  configuration rollback (rollback point-based), 234
  configuration save, 233, 233, 238
  data entry retrieval (interface table), 231
  data filtering, 239
  data filtering (conditional match), 243
  data filtering (regex match), 242
  edit-config operation, 228
  event notification subscription, 221, 222
  FIPS compliance, 219
  get/get-bulk operation, 226
  get-config/get-bulk-config operation, 227
  message format, 217
  NETCONF logging enable, 220
  NETCONF over SSH enable, 220
  over SOAP, 217
  over SOAP enable, 219
  parameter value change, 232
  protocols and standards, 218
  save-point/begin operation, 234
  save-point/commit operation, 235
  save-point/end operation, 236
  save-point/get-commit-information operation, 237
  save-point/get-commits operation, 236
  save-point/rollback operation, 235
  service operations, 225
  session establishment, 220
  session information retrieval, 246
  session termination, 247
  supported operations, 260
  XML view, 220
network
information center log minimum storage period, 59
Network Configuration Protocol. Use NETCONF
Network Time Protocol. Use NTP
NMM feature module debug, 6
NMM information center diagnostic log save to file, 57
NMM information center duplicate log suppression, 59
NMM information center interface link up/link down log generation, 60
NMM information center security log file management, 57
NMM information center security log save to file, 56
NMM information center synchronous log output, 59
NMM information center system log types, 48
NMM information center trace log file max size, 58
NMM local port mirroring configuration, 149
NMM local port mirroring group monitor port, 151
NMM local port mirroring group monitor port (in interface view), 151
NMM local port mirroring group monitor port (in system view), 151
NMM local port mirroring group source port, 150
NMM local port mirroring group source port (in interface view), 150
NMM local port mirroring group source port (in system view), 150
NMM NQA client DHCP operation, 94
NMM NQA client DLSw operation, 104
NMM NQA client DNS operation, 95
NMM NQA client FTP operation, 96
NMM NQA client history record save, 111
NMM NQA client HTTP operation, 97
NMM NQA client ICMP echo operation, 93
NMM NQA client operation, 93
NMM NQA client operation optional parameters, 106
NMM NQA client operation scheduling, 112
NMM NQA client path jitter operation, 105
NMM NQA client SNMP operation, 99
NMM NQA client statistics collection, 111
NMM NQA client TCP operation, 100
NMM NQA client template, 112
NMM NQA client threshold monitoring, 108
NMM NQA client UDP echo operation, 100
NMM NQA client UDP jitter operation, 98
NMM NQA client voice operation, 102
NMM NQA client+Track collaboration function, 107
NMM NQA DLSw operation configuration, 136
NMM NQA FTP operation configuration, 124
NMM NQA ICMP echo operation configuration, 120
NMM NQA path jitter operation configuration, 137
NMM NQA server, 92
NMM NQA TCP operation configuration, 130
NMM NQA UDP echo operation configuration, 131
NMM NQA voice operation configuration, 134
NMM NTP access control rights, 15
NMM NTP architecture, 8
NMM NTP association mode, 12
NMM NTP authentication, 16
NMM NTP broadcast association mode, 14
NMM NTP broadcast mode authentication, 20
NMM NTP client/server association mode, 12
NMM NTP client/server mode authentication, 16
NMM NTP local clock as reference source, 26
NMM NTP max number dynamic associations, 25
NMM NTP message receiving disable, 25
NMM NTP message source interface, 24
NMM NTP multicast association mode, 14
NMM NTP multicast mode authentication, 22
NMM NTP optional parameter configuration, 24
NMM NTP packet DSCP value, 26
NMM NTP symmetric active/passive association mode, 13
NMM NTP symmetric active/passive mode authentication, 18
NMM ping connectivity test, 1
NMM port mirroring remote destination group, 153
NMM port mirroring remote source group, 155
NMM port mirroring remote source group egress port, 156
NMM port mirroring remote group remote probe VLAN, 156
NMM port mirroring remote group source ports, 155
NMM SNMPv1 basics, 67
NMM SNMPv2c basics, 67
NMM SNMPv3 basics, 68
NMM SNTP authentication, 45
NMM SNTP NTP server specification, 44
NMM traceroute node failure identification, 4, 4
NMM traffic mirroring match criteria, 162
NMM traffic mirroring QoS policy, 163
NMM traffic mirroring QoS policy application, 163
NMM traffic mirroring QoS policy application (control plane), 164
NMM traffic mirroring QoS policy application (global), 164
NMM traffic mirroring QoS policy application (interface), 163
NMM traffic mirroring QoS policy application (VLAN), 164
NMM traffic mirroring traffic behavior, 163
NQA UDP template configuration, 143
NQA UDP tracert operation configuration, 132
quality analyzer. See NQA
sFlow counter sampling configuration, 169
sFlow flow sampling configuration, 168
network management
CWMP basic functions, 187
CWMP configuration, 187, 190, 196
EAA configuration, 176, 183
local port mirroring configuration with multiple monitor ports, 160
NMM information center configuration, 48, 53, 61
NMM information center log output (console), 61
NMM information center log output (Linux log host), 63
NMM information center log output (UNIX log host), 62
NMM IPv6 NTP client/server association mode, 28
NMM IPv6 NTP multicast association mode, 36
NMM IPv6 NTP symmetric active/passive association mode, 30
NMM Layer 2 remote port mirroring configuration, 153, 158
NMM local port mirroring configuration, 157
NMM NETCONF configuration, 216
NMM NQA collaboration configuration, 139
NMM NQA configuration, 90, 92, 120
NMM NQA DHCP operation configuration, 121
NMM NQA DNS operation configuration, 122
NMM NQA HTTP operation configuration, 125
NMM NQA ICMP template configuration, 141
NMM NQA SNMP operation configuration, 128
NMM NQA UDP jitter operation configuration, 126
NMM NTP broadcast association mode, 31
NMM NTP broadcast mode with authentication, 40
NMM NTP client/server association mode, 27
NMM NTP client/server mode with authentication, 39
NMM NTP configuration, 7, 12
NMM NTP multicast association mode, 33
NMM NTP symmetric active/passive association mode, 29
NMM PoE configuration, 213
NMM port mirroring configuration, 147, 157
NMM RMON alarm function configuration, 88
NMM RMON configuration, 81
NMM RMON Ethernet statistics group configuration, 85
NMM RMON history group configuration, 86
NMM SNMPv1 configuration, 75
NMM SNMPv2c configuration, 75
NMM SNMPv3 configuration (RBAC mode), 78
NMM SNMPv3 configuration (VACM mode), 76
NMM SNTP configuration, 44, 46
NMM traffic mirroring configuration, 162, 162, 165
NTP configuration, 27
packet capture configuration, 249
PMM Linux network, 172
RMON configuration, 85
sFlow configuration, 167, 167, 169
SNMP configuration, 75
NMM
captured packet saving, 256
CWMP ACS attributes, 191
CWMP ACS attributes (default) (CLI), 193
CWMP ACS attributes (preferred), 191
CWMP ACS autoconnect parameters, 194
CWMP ACS HTTPS SSL client policy, 196
CWMP basic functions, 187
CWMP configuration, 187, 190, 196
CWMP CPE ACS authentication parameters, 193
CWMP CPE ACS connection interface, 194
CWMP CPE ACS provision code, 193
CWMP CPE attributes, 193
CWMP CPE NAT traversal, 195
CWMP framework, 187
CWMP settings display, 196
displaying information center, 60
displaying NQA, 119
displaying NTP, 26
displaying PoE, 213
displaying SNMP settings, 74
displaying SNTP, 46
displaying the contents in a packet file, 257
EAA configuration, 176, 183
EAA environment variable configuration (user-defined), 179
283
EAA event monitor, 176  
EAA event monitor policy element, 177  
EAA event monitor policy environment variable, 178  
EAA event source, 176  
EAA monitor policy, 176  
EAA monitor policy configuration, 179  
EAA monitor policy configuration (CLI), 180, 183  
EAA monitor policy configuration (CLI-defined with environment variables), 184  
EAA monitor policy configuration (Tcl), 181, 185  
EAA monitor policy suspension, 182  
EAA RTM, 176  
feature module debug, 6  
filtering packet data to display, 256  
information center configuration, 48, 53, 61  
information center diagnostic log save to file, 57  
information center duplicate log suppression, 59  
information center interface link up/link down log generation, 60  
information center log default output rules, 49  
information center log destinations, 49  
information center log formats, 50  
information center log levels, 48  
information center log minimum storage period, 59  
information center log output (console), 53, 61  
information center log output (Linux log host), 63  
information center log output (log buffer), 55  
information center log output (log host), 54  
information center log output (monitor terminal), 54  
information center log output (UNIX log host), 62  
information center log save to file, 55  
information center security log file management, 57  
information center security log management, 56  
information center security log save to file, 56  
information center synchronous log output, 59  
information center system log types, 48  
information center trace log file max size, 58  
IPv6 NTP client/server association mode configuration, 28  
IPv6 NTP multicast association mode configuration, 36  
IPv6 NTP symmetric active/passive association mode configuration, 30  
Layer 2 remote port mirroring configuration, 153, 158  
local port mirroring configuration, 149, 157  
local port mirroring group monitor port, 151  
local port mirroring group monitor port (in interface view), 151  
local port mirroring group monitor port (in system view), 151  
local port mirroring group source port, 150  
local port mirroring group source port (in interface view), 150  
local port mirroring group source port (in system view), 150  
local port mirroring local group, 150  
maintaining information center, 60  
maintaining PoE, 213  
max PI power, 209  
NETCONF capability exchange, 221  
NETCONF CLI operations, 244, 245  
NETCONF CLI return, 248  
NETCONF configuration, 216, 219  
NETCONF configuration data retrieval (all modules), 228  
NETCONF configuration data retrieval (Syslog module), 230  
NETCONF configuration load, 233  
NETCONF configuration lock/unlock, 223, 224  
NETCONF configuration rollback, 233  
NETCONF configuration save, 233  
NETCONF data entry retrieval (interface table), 231  
NETCONF data filtering, 239  
NETCONF edit-config operation, 228  
NETCONF event notification subscription, 221, 222  
NETCONF get/get-bulk operation, 226  
NETCONF get-config/get-bulk-config operation, 227  
NETCONF over SOAP enable, 219  
NETCONF parameter value change, 232  
NETCONF save-point/begin operation, 234  
NETCONF save-point/commit operation, 235  
NETCONF save-point/end operation, 236  
NETCONF save-point/get-commit-information operation, 237  
NETCONF save-point/get-commits operation, 236  
NETCONF save-point/rollback operation, 235  
NETCONF service operations, 225  
NETCONF session establishment, 220
NETCONF session information retrieval, 246
NETCONF session termination, 247
NETCONF supported operations, 260
NQA client DHCP operation, 94
NQA client DLSw operation, 104
NQA client DNS operation, 95
NQA client FTP operation, 96
NQA client FTP template, 117
NQA client history record save, 111
NQA client HTTP operation, 97
NQA client HTTP template, 116
NQA client ICMP echo operation, 93
NQA client ICMP operation, 93
NQA client operation optional parameters, 106
NQA client operation scheduling, 112
NQA client path jitter operation, 105
NQA client SNMP operation, 99
NQA client statistics collection, 111
NQA client TCP operation, 100
NQA client TCP template, 114
NQA client template, 112
NQA client template (UDP), 115
NQA client template optional parameters, 118
NQA client threshold monitoring, 108
NQA client UDP echo operation, 100
NQA client UDP jitter operation, 98
NQA client voice operation, 102
NQA client+Track collaboration function, 107
NQA collaboration configuration, 139
NQA configuration, 90, 92, 120
NQA DHCP operation configuration, 121
NQA DLSw operation configuration, 136
NQA DNS operation configuration, 122
NQA FTP operation configuration, 124
NQA FTP template configuration, 145
NQA HTTP operation configuration, 125
NQA HTTP template configuration, 144
NQA ICMP echo operation configuration, 120
NQA ICMP template configuration, 141, 142
NQA operation, 90
NQA path jitter operation configuration, 137
NQA server, 92
NQA SNMP operation configuration, 128
NQA TCP operation configuration, 130
NQA TCP template configuration, 143
NQA threshold monitoring, 91
NQA UDP echo operation configuration, 131
NQA UDP jitter operation configuration, 126
NQA UDP template configuration, 143
NQA UDP tracer operation configuration, 132
NQA voice operation configuration, 134
NQA+Track collaboration, 91
NTP access control rights, 15
NTP architecture, 8
NTP association mode, 12
NTP authentication configuration, 16
NTP broadcast association mode configuration, 14, 31
NTP broadcast mode authentication configuration, 20
NTP broadcast mode with authentication, 40
NTP client/server association mode configuration, 27
NTP client/server mode authentication configuration, 16
NTP client/server mode with authentication, 39
NTP configuration, 7, 12
NTP enable, 12
NTP local clock as reference source, 26
NTP max number dynamic associations, 25
NTP message receiving disable, 25
NTP message source interface specification, 24
NTP multicast association mode, 14
NTP multicast association mode configuration, 33
NTP multicast mode authentication configuration, 22
NTP optional parameter configuration, 24
NTP packet DSCP value configuration, 26
NTP protocols and standards, 11
NTP security, 10
NTP symmetric active/passive association mode configuration, 29
NTP symmetric active/passive mode authentication configuration, 18
packet capture configuration, 249, 256
PI configuration using PoE profile, 211
PI power management, 210
ping address reachability determination, 1
ping command, 1
ping connectivity test, 1
PoE configuration, 207, 207, 213
PoE monitoring, 211
PoE PD detection (nonstandard), 209
PoE profile application, 212
PoE profile configuration, 212
PoE PSE firmware upgrade, 212
PoE troubleshooting, 214
port mirroring classification, 148
port mirroring configuration, 147, 157
port mirroring implementation, 148
port mirroring remote destination group, 153
port mirroring remote source group, 155
PSE power monitoring, 211
RMON alarm function configuration, 88
RMON alarm group sample type, 83
RMON configuration, 81, 85
RMON Ethernet statistics group configuration, 85
RMON group, 81
RMON history group configuration, 86
RMON private alarm group sample type, 83
sFlow agent+collector information configuration, 168
sFlow configuration, 167, 167, 169
dsFlow counter sampling configuration, 169
dsFlow display, 169
dsFlow flow sampling configuration, 168
dsFlow protocols and standards, 167
SNMP basics, 67
SNMP framework, 65
SNMP Get operation, 66
SNMP host notification send, 73
SNMP logging configuration, 72
SNMP MIB, 65
SNMP MIB access control mode, 66
SNMP MIB role-based access control, 66
SNMP MIB view-based access control, 66
SNMP notification, 72
SNMP protocol versions, 66
SNMP view-based MIB access control, 65
SNMPv1 configuration, 75
SNMPv2c configuration, 75
SNMPv3 configuration (RBAC mode ), 78
SNMPv3 configuration (VACM mode ), 76
SNTP authentication, 45
SNTP configuration, 44, 46
SNTP enable, 44
SNTP NTP server specification, 44
system debugging, 1, 5
system information diagnostic log default output rules, 49
system information hidden log default output rules, 50
system information security log default output rules, 49
system information trace log default output rules, 50
system maintenance, 1
tracert, 3, 4
tracert node failure identification, 4
traffic mirroring configuration, 162, 162, 165
traffic mirroring match criteria, 162
traffic mirroring QoS policy, 163
traffic mirroring QoS policy application, 163
traffic mirroring QoS policy application (control plane), 164
traffic mirroring QoS policy application (global), 164
traffic mirroring QoS policy application (interface), 163
traffic mirroring traffic behavior, 163
troubleshooting sFlow, 171
troubleshooting sFlow remote collector cannot receive packets, 171

NMS
NMM RMON alarm function configuration, 88
NMM RMON configuration, 81
NMM RMON Ethernet statistics group configuration, 85
NMM RMON history group configuration, 86
NMM SNMP Get operation, 66
NMM SNMP Notification operation, 66
NMM SNMP protocol versions, 66
NMM SNMP Set operation, 66
RMON configuration, 85
notifying
NMM NETCONF event notification subscription, 221, 222
NMM SNMP notification, 72
NMM SNMP Notification operation, 66
SNMP configuration, 75
SNMP host notification send, 73

NQA
client DHCP operation, 94
client DLSw operation, 104
client DNS operation, 95
client DNS template, 113
client enable, 93
client FTP operation, 96
client FTP template, 117
client history record save, 111
client HTTP operation, 97
client HTTP template, 116
client ICMP echo operation, 93
client ICMP template, 112
client operation, 93
client operation optional parameters, 106
client operation scheduling, 112
client path jitter operation, 105
client SNMP operation, 99
client statistics collection, 111
client TCP operation, 100
client TCP template, 114
client template (UDP), 115
client template configuration, 112
client template optional parameters, 118
client threshold monitoring, 108
client UDP echo operation, 100
client UDP jitter operation, 98
client voice operation, 102
client+Track collaboration function, 107
collaboration configuration, 139
collaboration, 90, 92, 120
DHCP operation configuration, 121
displaying, 119
displaying, 119
DLSw operation configuration, 136
DNS operation configuration, 122
DNS template configuration, 142
FTP operation configuration, 124
FTP template configuration, 145
HTTP operation configuration, 125
HTTP template configuration, 144
ICMP echo operation configuration, 140
ICMP template configuration, 141
ICMP operation, 90
path jitter operation configuration, 137
server configuration, 92
SNMP operation configuration, 128
TCP operation configuration, 130
TCP template configuration, 143
threshold monitoring, 91
Track collaboration function, 91
UDP echo operation configuration, 131
UDP jitter operation configuration, 126
UDP template configuration, 143
UDP tracert operation configuration, 132
voice operation configuration, 134

NTP
access control, 10
access control rights configuration, 15
architecture, 8
association mode configuration, 12
authentication, 11
authentication configuration, 16
broadcast association mode, 9
broadcast association mode configuration, 14, 31
broadcast client configuration, 14
broadcast mode authentication configuration, 20
broadcast mode max number dynamic associations, 25
broadcast mode with authentication, 40
broadcast server configuration, 14
client/server association mode, 9
client/server association mode configuration, 12, 27
client/server mode authentication configuration, 16
client/server mode max number dynamic associations, 25
client/server mode with authentication, 39
configuration, 7, 12, 27
configuration restrictions, 11
displaying, 26
enable, 12
how it works, 7
IPv6 client/server association mode configuration, 28
IPv6 multicast association mode configuration, 36
IPv6 symmetric active/passive association mode configuration, 30
local clock as reference source, 26
message receiving disable, 25
message source interface specification, 24
multicast association mode, 9
multicast association mode configuration, 14, 33
multicast client configuration, 14
multicast mode authentication configuration, 22
multicast mode max number dynamic associations, 25
multicast server configuration, 15
optional parameter configuration, 24
packet DSCP value configuration, 26
protocols and standards, 11
security, 10
SNTP authentication, 45
SNTP configuration, 44, 46
SNTP server specification, 44
symmetric active/passive association mode, 9
symmetric active/passive association mode configuration, 13, 29
symmetric active/passive mode authentication configuration, 18
symmetric active/passive mode max number dynamic associations, 25

O
outbound
NMM port mirroring, 147
outputting
NMM information center log default output rules, 49
NMM information center logs to Linux log host, 63
NMM information center logs to log buffer, 55
NMM information center logs to UNIX log host, 62
NMM information center synchronous log output, 59
NMM information logs to console, 53, 61
NMM information logs to log host, 54
NMM information logs to monitor terminal, 54

P

packet
NMM NTP DSCP value configuration, 26
NMM port mirroring configuration, 147, 157
NMM SNTP configuration, 44, 46
NMM traffic mirroring configuration, 162, 162, 165
NMM traffic mirroring match criteria, 162
NMM traffic mirroring QoS policy, 163
NMM traffic mirroring QoS policy application, 163
NMM traffic mirroring QoS policy application (control plane), 164
NMM traffic mirroring QoS policy application (global), 164
NMM traffic mirroring QoS policy application (interface), 163
NMM traffic mirroring QoS policy application (VLAN), 164
NMM traffic mirroring traffic behavior, 163

packet capture
  capture filter expression, 254
  capture filter operator, 251
  configuration, 249, 257
  constant, 249
  display filter expression, 255
  display filter keyword, 252
  display filter operator, 253
  displaying the contents in a packet file, 257
  filtering packet data to display, 256
  packet capture configuration, 256
  saving captured packet, 256
  variable, 249

parameter
  NMM NETCONF parameter value change, 232
  NMM NQA client history record save, 111
  NMM NQA client operation optional parameters, 106

NMM NQA client template optional parameters, 118
NMM NTP local clock as reference source, 26
NMM NTP max number dynamic associations, 25
NMM NTP message receiving disable, 25
NMM NTP message source interface, 24
NMM NTP optional parameter configuration, 24
NMM SNMP basics, 67
NMM SNMPv1 basics, 67
NMM SNMPv2c basics, 67
NMM SNMPv3 basics, 68

path
  NMM NQA client path jitter operation, 105
  NMM NQA path jitter, 90
  NMM NQA path jitter operation configuration, 137

PD

detection mode, 209
PoE PD detection (nonstandard), 209

performing
  NETCONF save-point/begin operation, 234
  NETCONF save-point/commit operation, 235
  NETCONF save-point/end operation, 236
  NETCONF save-point/get-commit-information operation, 237
  NETCONF save-point/get-commits operation, 236
  NETCONF save-point/rollback operation, 235
  NMM NETCONF CLI operations, 244, 245
  NMM NETCONF edit-config operation, 228
  NMM NETCONF get/bulk operation, 226
  NMM NETCONF get-config/bulk-config operation, 227
  NMM NETCONF service operations, 225

PI

  NMM PoE enable, 208
  troubleshooting PoE profile, 215
  troubleshooting priority failure, 214

PI configuration
  PoE profile, 212

PI priority
  setting failure, 214

ping
  NMM address reachability determination, 1, 1
  NMM network connectivity test, 1
  NMM system maintenance, 1

PMM
  displaying, 172
  kernel thread deadloop detection, 173
  kernel thread monitoring, 173
  kernel thread starvation detection, 174
  Linux kernel thread, 172
Linux network, 172
Linux user, 172
maintaining, 172
user PMM display, 172

PoE
configuration, 207, 207, 213
displaying, 213
max PI power configuration, 209
monitoring configuring, 211
PD detection (nonstandard), 209
PD detection mode, 209
PI configuration using PoE profile, 211
PI power management, 210
PoE enabling for PI, 208
profile application, 212
profile configuration, 212
PSE firmware upgrade, 212
PSE power monitoring, 211
troubleshooting, 214
troubleshooting PI critical priority failure, 214
troubleshooting PoE profile failure, 215

PoE profile
application failure, 215

policy
CWMP ACS HTTPS SSL client policy, 196
EAA event monitor policy element, 177
EAA monitor policy configuration (Tcl), 181
NMM EAA configuration, 176, 183
NMM EAA environment variable configuration (user-defined), 179
NMM EAA event monitor policy environment variable, 178
NMM EAA monitor policy, 176
NMM EAA monitor policy configuration, 179
NMM EAA monitor policy configuration (CLI), 180, 183
NMM EAA monitor policy configuration (CLI-defined with environment variables), 184
NMM EAA monitor policy configuration (Tcl), 185
NMM EAA monitor policy suspension, 182
NMM traffic mirroring QoS policy, 163
NMM traffic mirroring QoS policy application, 163
NMM traffic mirroring QoS policy application (control plane), 164
NMM traffic mirroring QoS policy application (global), 164
NMM traffic mirroring QoS policy application (interface), 163

NMM traffic mirroring QoS policy application (VLAN), 164

port
configuring local mirroring to support multiple monitor ports, 151
mirroring. See port mirroring
NMM IPv6 NTP client/server association mode, 28
NMM IPv6 NTP multicast association mode, 36
NMM IPv6 NTP symmetric active/passive association mode, 30
NMM NTP association mode, 12
NMM NTP broadcast association mode, 31
NMM NTP broadcast mode with authentication, 40
NMM NTP client/server association mode, 27
NMM NTP client/server mode with authentication, 39
NMM NTP configuration, 7, 12
NMM NTP multicast association mode, 33
NMM NTP symmetric active/passive association mode, 29
NMM SNTP configuration, 44, 46

port mirroring
classification, 148
configuration, 147, 157
configuring local mirroring to support multiple monitor ports, 151
destination, 147
direction (bidirectional), 147
direction (inbound), 147
direction (outbound), 147
displaying, 157
egress port, 147
implementation, 148
Layer 2 remote port mirroring configuration, 158
local, 148
local configuration, 149
local group creation, 150
local group monitor port, 151
local group monitor port (in interface view), 151
local group monitor port (in system view), 151
local group monitor port configuration restrictions, 151
local group source port, 150
local group source port (in interface view), 150
local group source port (in system view), 150
local group source port configuration restrictions, 150
local mirroring configuration, 157
local mirroring configuration with multiple monitor ports, 160
mirroring group, 147
monitor port to remote probe VLAN assignment, 155
NMM Layer 2 remote configuration, 153
reflector port, 147
remote, 148
remote destination group configuration, 153
remote destination group creation, 153
remote destination group monitor port, 154
remote destination group remote probe VLAN, 154
remote probe VLAN, 147
remote source group configuration, 155
source, 147
terminology, 147

term
power
NMM PoE configuration, 213
over Ethernet. Use PoE sourcing equipment. Use PSE
power management
PI configuration, 210
power monitoring
PSE configuration, 211
private
NMM RMON private alarm group, 82
NMM RMON private alarm group sample type, 83
procedure
applying NMM traffic mirroring QoS policy, 163
applying NMM traffic mirroring QoS policy (control plane), 164
applying NMM traffic mirroring QoS policy (global), 164
applying NMM traffic mirroring QoS policy (interface), 163
applying NMM traffic mirroring QoS policy (VLAN), 164
applying PoE profile, 212
assigning CWMP ACS attribute (preferred)(CLI), 192
assigning CWMP ACS attribute (preferred)(DHCP server), 191
changing NMM NETCONF parameter value, 232
configuring CWMP, 190, 196
configuring CWMP ACS attribute, 191
configuring CWMP ACS attribute (default)(CLI), 193
configuring CWMP ACS attribute (preferred), 191
configuring CWMP ACS autoconnect parameters, 194
configuring CWMP ACS close-wait timer, 195
configuring CWMP ACS connection retry max number, 195
configuring CWMP ACS periodic Inform feature, 194
configuring CWMP CPE ACS authentication parameters, 193
configuring CWMP CPE ACS connection interface, 194
configuring CWMP CPE ACS provision code, 193
configuring CWMP CPE attribute, 193
configuring CWMP CPE NAT traversal, 195
configuring EAA monitor policy (Tcl), 181
configuring information center log suppression, 60
configuring local mirroring to support multiple monitor ports, 151
configuring local port mirroring with multiple monitor ports, 160
configuring max PI power, 209
configuring NMM EAA environment variable (user-defined), 179
configuring NMM EAA monitor policy, 179
configuring NMM EAA monitor policy (CLI), 180, 183
configuring NMM EAA monitor policy (CLI-defined with environment variables), 184
configuring NMM EAA monitor policy (Tcl), 185
configuring NMM information center, 53, 61
configuring NMM information center trace log file max size, 58
configuring NMM IPv6 NTP client/server association mode, 28
configuring NMM IPv6 NTP multicast association mode, 36
configuring NMM IPv6 NTP symmetric active/passive association mode, 30
configuring NMM Layer 2 remote port mirroring, 153, 158
configuring NMM local port mirroring, 149, 157
configuring NMM local port mirroring group monitor port, 151
configuring NMM local port mirroring group monitor port (in interface view), 151
configuring NMM local port mirroring group monitor port (in system view), 151
configuring NMM local port mirroring group source ports, 150
configuring NMM local port mirroring group source ports (in interface view), 150
configuring NMM local port mirroring group source ports (in system view), 150
configuring NMM NETCONF, 219
configuring NMM NQA, 92, 120
configuring NMM NQA client DHCP operation, 94
configuring NMM NQA client DLSw operation, 104
configuring NMM NQA client DNS operation, 95
configuring NMM NQA client DNS template, 113
configuring NMM NQA client FTP operation, 96
configuring NMM NQA client FTP template, 117
configuring NMM NQA client history record save, 111
configuring NMM NQA client HTTP operation, 97
configuring NMM NQA client HTTP template, 116
configuring NMM NQA client ICMP echo operation, 93
configuring NMM NQA client ICMP template, 112
configuring NMM NQA client operation, 93
configuring NMM NQA client operation optional parameters, 106
configuring NMM NQA client path jitter operation, 105
configuring NMM NQA client SNMP operation, 99
configuring NMM NQA client statistics collection, 111
configuring NMM NQA client TCP operation, 100
configuring NMM NQA client TCP template, 114
configuring NMM NQA client template, 112
configuring NMM NQA client template optional parameters, 118
configuring NMM NQA client threshold monitoring, 108
configuring NMM NQA client UDP echo operation, 100
configuring NMM NQA client UDP jitter operation, 98
configuring NMM NQA client voice operation, 102
configuring NMM NQA client+Track collaboration function, 107
configuring NMM NQA collaboration configuration, 139
configuring NMM NQA DHCP operation, 121
configuring NMM NQA DLSw operation configuration, 136
configuring NMM NQA DNS operation, 122
configuring NMM NQA DNS template, 142
configuring NMM NQA FTP operation configuration, 124
configuring NMM NQA FTP template, 145
configuring NMM NQA HTTP operation, 125
configuring NMM NQA HTTP template, 144
configuring NMM NQA ICMP echo operation, 120
configuring NMM NQA ICMP template, 141
configuring NMM NQA path jitter operation configuration, 137
configuring NMM NQA server, 92
configuring NMM NQA SNMP operation, 128
configuring NMM NQA TCP configuration, 130
configuring NMM NQA TCP template, 143
configuring NMM NQA UDP echo operation, 131
configuring NMM NQA UDP jitter operation, 126
configuring NMM NQA voice operation configuration, 134
configuring NMM NTP, 12
configuring NMM NTP access control rights, 15
configuring NMM NTP association mode, 12
configuring NMM NTP authentication, 16
configuring NMM NTP broadcast association mode, 14, 31
configuring NMM NTP broadcast client, 14
configuring NMM NTP broadcast mode authentication, 20
configuring NMM NTP broadcast mode with authentication, 40
configuring NMM NTP broadcast server, 14
configuring NMM NTP client/server association mode, 12, 27
configuring NMM NTP client/server mode authentication, 16
configuring NMM NTP client/server mode with authentication, 39
configuring NMM NTP local clock as reference source, 26
configuring NMM NTP max number dynamic associations, 25
configuring NMM NTP multicast association mode, 14, 33
configuring NMM NTP multicast client, 14
configuring NMM NTP multicast mode authentication, 22
configuring NMM NTP multicast server, 15
configuring NMM NTP optional parameters, 24
configuring NMM NTP packet DSCP value, 26
configuring NMM NTP symmetric active/passive association mode, 13, 29
configuring NMM NTP symmetric active/passive mode authentication, 18
configuring NMM PoE, 213
configuring NMM port mirroring monitor port to remote probe VLAN assignment, 155
configuring NMM port mirroring remote destination group monitor port, 154
configuring NMM port mirroring remote destination group on the destination device, 153
configuring NMM port mirroring remote destination group remote probe VLAN assignment, 155
configuring NMM port mirroring remote destination group on the destination device, 153
configuring NMM port mirroring remote destination group remote probe VLAN, 156
configuring NMM port mirroring remote destination group source ports, 155
configuring NMM RMON alarm function, 84, 88, 88
configuring NMM RMON Ethernet statistics group, 85, 85
configuring NMM RMON history group, 86, 86
configuring NMM RMON statistics function, 83
configuring NMM SNMP basic parameters, 67
configuring NMM SNMP logging, 72
configuring NMM SNMP notification, 72
configuring NMM SNMPv1, 75
configuring NMM SNMPv1 basics, 67
configuring NMM SNMPv2c, 75
configuring NMM SNMPv2c basics, 67
configuring NMM SNMPv3 (RBAC mode ), 78
configuring NMM SNMPv3 (VACM mode ), 76
configuring NMM SNMPv3 basic parameters, 68
configuring NMM SNTP, 44, 46
configuring NMM SNTP authentication, 45
configuring NMM traffic mirroring, 162, 165
configuring NMM traffic mirroring match criteria, 162
configuring NMM traffic mirroring QoS policy, 163
configuring NMM traffic mirroring traffic behavior, 163
configuring NQA client template (UDP), 115
configuring NQA UDP template, 143
configuring NQA UDP tracer operation, 132
configuring packet capture, 256, 257
configuring PD detection mode, 209
configuring PI power management, 210
configuring PI using PoE profile, 211
configuring PMM kernel thread deadloop detection, 173
configuring PMM kernel thread starvation detection, 174
configuring PoE, 207
configuring PoE monitoring, 211
configuring PoE profile, 212
configuring PSE power monitoring, 211
configuring sFlow, 167, 169
configuring sFlow agent+collector information, 168
configuring sFlow counter sampling, 169
configuring sFlow flow sampling, 168
configuring SNMPv1 host notification send, 73
configuring SNMPv2c host notification send, 73
configuring SNMPv3 host notification send, 73
creating NMM local port mirroring group, 150
creating NMM port mirroring remote destination group, 153
creating NMM port mirroring remote source group, 155
creating NMM port mirroring remote source group source ports, 155
creating NMM RMON Ethernet statistics entry, 83
creating NMM RMON history control entry, 83
debugging NMM feature module, 6
determining NMM address reachability with ping, 1
disabling NMM information center interface link up/link down log generation, 60
disabling NMM NTP message interface receiving, 25
displaying CWMP settings, 196
displaying NMM EAA settings, 182
displaying NMM information center, 60
displaying NMM NQA, 119
displaying NMM NTP, 26
displaying NMM PoE, 213
displaying NMM port mirroring, 157
displaying NMM RMON settings, 85
displaying NMM sFlow, 169
displaying NMM SNTP, 46
displaying PMM, 172
displaying PMM kernel threads, 174
displaying the contents in a packet file), 257
displaying user PMM, 172
enabling CWMP, 191
enabling NETCONF logging, 220
enabling NETCONF over SSH, 220
enabling NMM information center duplicate log suppression, 59
enabling NMM information center synchronous log information output, 59
enabling CWMP, 191
enabling NETCONF logging, 220
enabling NETCONF over SSH, 220
enabling NMM information center duplicate log suppression, 59
enabling NMM information center synchronous log information output, 59
enabling NMM NETCONF over SOAP, 219
enabling NMM NQA client, 93
enabling NMM NTP, 12
enabling NMM PoE PI, 208
enabling NMM SNMP notification, 72
enabling NMM SNTP, 44
enabling PoE PD detection (nonstandard), 209
entering NMM NETCONF XMLview, 220
establishing NMM NETCONF session, 220
exchanging NMM NETCONF capabilities, 221
filtering NMM NETCONF data, 239
filtering NMM NETCONF data (conditional match), 243
filtering NMM NETCONF data (regex match), 242
filtering packet data to display, 256
identifying NMM node failure with tracert, 4, 4
loading NMM NETCONF configuration, 233, 238
locking NMM NETCONF configuration, 223, 224
maintaining NMM information center, 60
maintaining NMM PoE, 213
maintaining PMM, 172
maintaining PMM kernel threads, 174
managing NMM information center security log, 56
managing NMM information center security log file, 57
monitoring kernel thread (PMM), 173
outputting NMM information center logs to console, 53, 61
outputting NMM information center logs to Linux log host, 63
outputting NMM information center logs to log buffer, 55
outputting NMM information center logs to log host, 54
outputting NMM information center logs to monitor terminal, 54
outputting NMM information center logs to UNIX log host, 62
performing NETCONF save-point/begin operation, 234
performing NETCONF save-point/commit operation, 235
performing NETCONF save-point/end operation, 236
performing NETCONF save-point/get-commit-information operation, 237
performing NETCONF save-point/get-commits operation, 236
performing NETCONF save-point/rollback operation, 235
performing NMM NETCONF CLI operations, 244, 245
performing NMM NETCONF edit-config operation, 228
performing NMM NETCONF get/get-bulk operation, 226
performing NMM NETCONF get-config/get-bulk-config operation, 227
performing NMM NETCONF service operations, 225
retrieving NETCONF data entry (interface table), 231
retrieving NMM NETCONF configuration data (all modules), 228
retrieving NMM NETCONF configuration data (Syslog module), 230
retrieving NMM NETCONF session information, 246
returning to NMM NETCONF CLI, 248
rolling back NETCONF configuration (rollback point-based), 234
rolling back NMM NETCONF configuration, 233
rolling back NMM NETCONF configuration (configuration file-based), 233
saving captured packet, 256
saving captured packets to a file, 258
saving NMM information center diagnostic logs to file, 57
saving NMM information center log to log file, 55
saving NMM information center security logs to security log file, 56
saving NMM NETCONF configuration, 233, 233, 238
scheduling CWMP ACS connection initiation, 195
scheduling NMM NQA client operation, 112
setting information center log minimum storage period, 59
specifying CWMP ACS HTTPS SSL client policy, 196
specifying NMM NTP message source interface, 24
specifying NMM SNTP NTP server, 44
subscribing to NMM NETCONF event notifications, 221, 222
suspending NMM EAA monitor policy, 182
terminating NMM NETCONF session, 247
testing NMM network connectivity with ping, 1
troubleshooting sFlow remote collector cannot receive packets, 171
unlocking NMM NETCONF configuration, 223, 224
upgrading PSE firmware, 212

process monitoring and maintenance. See PMM profile

PI configuration using PoE profile, 211
PoE profile application, 212
PoE profile configuration, 212

protocol
capture filter expression, 254
display filter expression, 255
display filter keyword, 252

protocols and standards
NMM NETCONF, 216, 218
NMM NTP, 11
NMM RMON, 83
NMM SNMP versions, 66
sFlow, 167
SNMP configuration, 75

provision code (ACS), 193

PSE
PoE configuration, 207, 207

Q
QoS
NMM traffic mirroring configuration, 162, 162, 165
NMM traffic mirroring match criteria, 162
NMM traffic mirroring QoS policy, 163
NMM traffic mirroring QoS policy application, 163
NMM traffic mirroring QoS policy application (control plane), 164
NMM traffic mirroring QoS policy application (global), 164
NMM traffic mirroring QoS policy application (interface), 163
NMM traffic mirroring QoS policy application (VLAN), 164
NMM traffic mirroring traffic behavior, 163

querying
NMM NTP query ACL, 10

R
real-time event manager. See RTM
reflector port
NMM Layer 2 remote port mirroring, 147

regex match
NETCONF data filtering (column-based), 240
regex match NETCONF data filtering, 242

regular expression. Use regex remote
NMM Layer 2 remote port mirroring, 153
NMM remote port mirroring, 148
NMM remote port mirroring destination group, 153
NMM remote port mirroring destination group creation, 153
NMM remote port mirroring destination group monitor port, 154
NMM remote port mirroring destination group remote probe VLAN, 154
NMM remote port mirroring monitor port to remote probe VLAN assignment, 155
NMM remote port mirroring source group, 155
NMM remote port mirroring source group creation, 155
NMM remote port mirroring source group egress port, 156
NMM remote port mirroring source group remote probe VLAN, 156
NMM remote port mirroring source group source ports, 155

Remote Network Monitoring. Use RMON remote probe VLAN
NMM Layer 2 remote port mirroring, 147
NMM port mirroring monitor port to remote probe VLAN assignment, 155
NMM port mirroring remote destination group, 154
NMM port mirroring remote source group remote probe VLAN, 156

restrictions
NMM local port mirroring group monitor port configuration, 151
NMM local port mirroring group source port configuration, 150
NTP configuration, 11
SNTP configuration, 11

retrieving
NETCONF data entry (interface table), 231
NMM NETCONF configuration data (all modules), 228
NMM NETCONF configuration data (Syslog module), 230
NMM NETCONF session information, 246

returning
NMM NETCONF CLI return, 248

RMON
alarm function configuration, 84, 88
alarm group, 82
alarm group sample type, 83
configuration, 81, 85
displaying settings, 85
Ethernet statistics entry creation, 83
Ethernet statistics group, 81
Ethernet statistics group configuration, 85
event group, 81
history control entry creation, 83
history group, 81
history group configuration, 86
private alarm group, 82
private alarm group sample type, 83
protocols and standards, 83
statistics function configuration, 83
role-based MIB access control, 66
rollback
NETCONF save-point/begin operation, 234
NETCONF save-point/commit operation, 235
NETCONF save-point/end operation, 236
NETCONF save-point/get-commit-information operation, 237
NETCONF save-point/get-commits operation, 236
NETCONF save-point/rollback operation, 235
routing
NMM IPv6 NTP client/server association mode, 28
NMM IPv6 NTP multicast association mode, 36
NMM IPv6 NTP symmetric active/passive association mode, 30
NMM NTP association mode, 12
NMM NTP broadcast association mode, 31
NMM NTP broadcast mode with authentication, 40
NMM NTP client/server association mode, 27
NMM NTP client/server mode with authentication, 39
NMM NTP configuration, 7, 12
NMM NTP multicast association mode, 33
NMM NTP symmetric active/passive association mode, 29
NMM SNTP configuration, 44, 46
RTM
NMM EAA, 176
NMM EAA configuration, 176, 183
NMM information center log default output rules, 49
NMM system information default diagnostic log output rules, 49
NMM system information default hidden log output, 50
NMM system information default security log output, 49
NMM system information default trace log output, 50
runtime
NMM EAA event monitor policy runtime, 178
S
sampling
NMM RMON alarm group sample type, 83
NMM RMON private alarm group sample type, 83
Sampled Flow. Use sFlow sFlow counter sampling, 169
sFlow flow sampling configuration, 168
saving
captured packet, 256
NMM information center diagnostic logs to file, 57
NMM information center log to log file, 55
NMM information center security logs to file, 56
NMM NETCONF configuration, 233, 233, 238
NMM NQA client history records, 111
saving captured packets to a file, 258
scheduling
CWMP ACS connection initiation, 195
NMM NQA client operation, 112
security
NMM information center security log file management, 57
NMM information center security log management, 56
NMM information center security log save to file, 56
NMM information center security logs, 48
NMM NTP, 10
NMM NTP access control rights, 15
NMM NTP authentication, 11, 16
NMM NTP broadcast mode authentication, 20
NMM NTP client/server mode authentication, 16
NMM NTP multicast mode authentication, 22
NMM NTP symmetric active/passive mode authentication, 18
NMM SNTP authentication, 45
server
NMM NQA, 92
NMM NTP broadcast server configuration, 14
NMM NTP multicast server configuration, 15
NMM SNTP configuration, 44, 46
NMM SNTP NTP server specification, 44

service
NMM NETCONF configuration data retrieval (all modules), 228
NMM NETCONF configuration data retrieval (Syslog module), 230
NMM NETCONF configuration load, 233
NMM NETCONF configuration rollback, 233
NMM NETCONF configuration save, 233
NMM NETCONF edit-config operation, 228
NMM NETCONF get/get-bulk operation, 226
NMM NETCONF get-config/get-bulk-config operation, 227
NMM NETCONF operations, 225
NMM NETCONF parameter value change, 232

session
NMM NETCONF session establishment, 220
NMM NETCONF session information retrieval, 246
NMM NETCONF session termination, 247

Set operation (SNMP), 66, 72

setting
information center log minimum storage period, 59
severity level (system information), 48

sFlow
agent+collector information configuration, 168
counter sampling configuration, 169
display, 169
flow sampling configuration, 168
protocols and standards, 167
troubleshoot, 171
troubleshoot remote collector cannot receive packets, 171

Use SNMP
Simplified NTP. See SNTP
SNMP
agent, 65
agent notification, 72
basic parameter configuration, 67
displaying settings, 74
FIPS compliance, 67
framework, 65
Get operation, 66, 72
host notification send, 73
logging configuration, 72
manager, 65

MIB, 65, 65
MIB access control mode, 66
MIB view-based access control, 65, 66, 66
NMM NQA, 90
NMM NQA client SNMP operation, 99
NMM NQA SNMP operation configuration, 128
NMM RMON alarm function configuration, 88
NMM RMON configuration, 81
NMM RMON Ethernet statistics group configuration, 85
NMM RMON history group configuration, 86
notification configuration, 72
notification enable, 72
Notification operation, 66
protocol versions, 66
RMON configuration, 85
Set operation, 66, 72
SNMPv1 basic parameter configuration, 67
SNMPv1 configuration, 75
SNMPv2c basic parameter configuration, 67
SNMPv2c configuration, 75
SNMPv3 basic parameter configuration, 68
SNMPv3 configuration (RBAC mode ), 78
SNMPv3 configuration (VACM mode ), 76

SNMPv1
basic parameter configuration, 67
configuration, 75
host notification send, 73
Notification operation, 66
protocol version, 66

SNMPv2c
basic parameter configuration, 67
configuration, 75
host notification send, 73
Notification operation, 66
protocol version, 66

SNMPv3
basic parameter configuration, 68
configuration (RBAC mode ), 78
configuration (VACM mode ), 76
Notification operation, 66
notification send, 73
protocol version, 66

SNTP
authentication, 45
configuration, 44, 46
configuration restrictions, 11, 44
displaying, 46
enable, 44
NTP server specification, 44
SOAP
  NMM NETCONF message format, 217
  NMM NETCONF over SOAP enable, 219
source
  NMM port mirroring, 147
specifying
  CWMP ACS HTTPS SSL client policy, 196
  NMM NTP message source interface, 24
  NMM SNTP NTP server, 44
SSH
  NETCONF over SSH enable, 220
SSL
  CWMP ACS HTTPS SSL client policy, 196
starvation detection (Linux kernel thread PMM), 174
statistics
  NMM NQA client statistics collection, 111
  NMM RMON alarm function configuration, 88
  NMM RMON configuration, 81
  NMM RMON Ethernet statistics entry, 83
  NMM RMON Ethernet statistics group, 81
  NMM RMON Ethernet statistics group configuration, 85, 85
  NMM RMON history control entry, 83
  NMM RMON history group configuration, 86
  NMM RMON statistics function, 83
  RMON configuration, 85
  sFlow agent+collector information configuration, 168
  sFlow configuration, 167, 167, 169
  sFlow counter sampling configuration, 169
  sFlow flow sampling configuration, 168
subscribing
  NMM NETCONF event notification, 221, 222
suppressing
  NMM information center duplicate log suppression, 59
suspending
  NMM EAA monitor policy, 182
switch
  NMM module debug, 6
  NMM screen output, 6
symmetric
  NMM IPv6 NTP symmetric active/passive association mode, 30
  NMM NTP symmetric active/passive association mode, 9, 13, 18, 29
  NMM NTP symmetric active/passive mode max number dynamic associations, 25
synchronizing
  NMM information center synchronous log output, 59
  NMM NTP configuration, 7, 12
  NMM NTP synchronization ACL, 10
  NMM SNTP configuration, 44, 46
Syslog
  NMM NETCONF configuration data retrieval (Syslog module), 230
system
  NMM diagnostic log default output rules, 49
  NMM hidden log default output rules, 50
  NMM information center duplicate log suppression, 59
  NMM information center interface link up/link down log generation, 60
  NMM information center log destinations, 49
  NMM information center log levels, 48
  NMM information center log output (console), 53, 61
  NMM information center log output (Linux log host), 63
  NMM information center log output (log buffer), 55
  NMM information center log output (log host), 54
  NMM information center log output (monitor terminal), 54
  NMM information center log output (UNIX log host), 62
  NMM information center log save to file, 55
  NMM information center log types, 48
  NMM information center security log file management, 57
  NMM information center security log management, 56
  NMM information center security log save to file, 56
  NMM information center synchronous log output, 59
  NMM information log formats, 50
  NMM log default output rules, 49
  NMM security log default output rules, 49
  NMM trace log default output rules, 50
system administration
  NMM debugging, 1
  NMM feature module debug, 6
  NMM ping, 1
  NMM ping address reachability, 1
  NMM ping command, 1
  NMM ping connectivity test, 1
  NMM system debugging, 5
  NMM tracert, 1, 3, 4
  NMM tracert node failure identification, 4
system information
NMM information center configuration, 48, 53, 61

T

table
  NETCONF data entry retrieval (interface table), 231

Tcl
  EAA monitor policy configuration, 181
  NMM EAA configuration, 176, 183
  NMM EAA monitor policy configuration, 185

TCP
  NMM NQA, 90
  NMM NQA client TCP operation, 100
  NMM NQA client TCP template, 114
  NMM NQA TCP operation configuration, 130
  NMM NQA TCP template configuration, 143

template
  NMM NQA client DNS template, 113
  NMM NQA client FTP template, 117
  NMM NQA client HTTP template, 116
  NMM NQA client ICMP template, 112
  NMM NQA client TCP template, 114
  NMM NQA client template, 112
  NMM NQA client template optional parameters, 118
  NMM NQA DNS template configuration, 142
  NMM NQA FTP template configuration, 145
  NMM NQA HTTP template configuration, 144
  NMM NQA ICMP template configuration, 141
  NMM NQA TCP template configuration, 143
  NQA client template (UDP), 115
  NQA UDP template configuration, 143

terminating
  NMM NETCONF session, 247

testing
  NMM ping network connectivity test, 1

threshold
  NMM NQA client threshold monitoring, 91, 108
  NMM NQA operation reaction entry, 108
  NMM NQA operation support accumulate type, 108
  NMM NQA operation support average type, 108
  NMM NQA operation support consecutive type, 108
  NMM NQA operation triggered action none, 108
  NMM NQA operation triggered action trap-only, 108
  NMM NQA operation triggered action trigger-only, 108

time
  NMM NTP configuration, 7, 12
  NMM NTP local clock as reference source, 26
  NMM SNTP configuration, 44, 46

timer
  CWMP ACS close-wait timer, 195

traceroute. See tracert

tracert
  NMM IP address retrieval, 3
  NMM node failure detection, 3
  NMM node failure identification, 4
  NMM system maintenance, 1
  node failure detection, 4
  NQA UDP tracert operation configuration, 132

tracing
  information center log minimum storage period, 59
  NMM information center trace log file max size, 58

Track
  NMM NQA client+Track collaboration function, 107
  NMM NQA collaboration, 91
  NMM NQA collaboration configuration, 139

traffic
  mirroring. See traffic mirroring
  NMM RMON alarm function configuration, 88
  NMM RMON configuration, 81
  NMM RMON Ethernet statistics group configuration, 85
  NMM RMON history group configuration, 86
  RMON configuration, 85
  sFlow agent+collector information configuration, 168
  sFlow configuration, 167, 167, 169
  sFlow counter sampling configuration, 169
  sFlow flow sampling configuration, 168

traffic mirroring
  configuration, 162, 162, 165
  match criteria configuration, 162
  QoS policy application, 163
  QoS policy application (control plane), 164
  QoS policy application (global), 164
  QoS policy application (interface), 163
  QoS policy application (VLAN), 164
  QoS policy configuration, 163
  traffic behavior configuration, 163

trapping
  NMM SNMP notification, 72

triggering
NMM NQA operation threshold triggered action none, 108
NMM NQA operation threshold triggered action trap-only, 108
NMM NQA operation threshold triggered action trigger-only, 108

troubleshooting
PI critical priority failure, 214
PoE, 214
PoE profile failure, 215
sFlow, 171
sFlow remote collector cannot receive packets, 171

UDP
NMM IPv6 NTP client/server association mode, 28
NMM IPv6 NTP multicast association mode, 36
NMM IPv6 NTP symmetric active/passive association mode, 30
NMM NQA, 90
NMM NQA client UDP echo operation, 100
NMM NQA client UDP jitter operation, 98
NMM NQA UDP echo operation configuration, 131
NMM NQA UDP jitter operation configuration, 126
NMM NTP association mode, 12
NMM NTP broadcast association mode, 31
NMM NTP broadcast mode with authentication, 40
NMM NTP client/server association mode, 27
NMM NTP client/server mode with authentication, 39
NMM NTP configuration, 7, 12
NMM NTP multicast association mode, 33
NMM NTP symmetric active/passive association mode, 29
NQA client template, 115
NQA template configuration, 143
NQA UDP tracert operation configuration, 132
sFlow configuration, 167, 167, 169

UNIX
NMM information center log host, 62
unlocking
NMM NETCONF configuration, 223, 224
upgrading
PoE PSE firmware, 212
user
PMM Linux user, 172

V

value
NMM NETCONF parameter value change, 232
variable
capture filter keyword, 249
NMM EAA environment variable configuration (user-defined), 179
NMM EAA event monitor policy environment system-defined, 178
NMM EAA event monitor policy environment system-defined (event-specific), 178
NMM EAA event monitor policy environment system-defined (public), 178
NMM EAA event monitor policy environment user-defined, 179
NMM EAA event monitor policy environment variable, 178

view-based MIB access control, 65, 66

VLAN
capture filter expression, 254
NMM Layer 2 remote port mirroring configuration, 153
NMM local port mirroring configuration, 149
NMM local port mirroring group monitor port, 151
NMM local port mirroring group monitor port (in interface view), 151
NMM local port mirroring group monitor port (in system view), 151
NMM local port mirroring group source port, 150
NMM local port mirroring group source port (in interface view), 150
NMM local port mirroring group source port (in system view), 150
NMM port mirroring configuration, 147, 157
NMM port mirroring remote probe VLAN, 147
NMM port mirroring remote source group remote probe VLAN, 156
NMM traffic mirroring configuration, 162, 162, 165
NMM traffic mirroring QoS policy application, 164

voice
NMM NQA, 90
NMM NQA client voice operation, 102
NMM NQA voice operation configuration, 134

X

XML
NMM NETCONF capability exchange, 221
NMM NETCONF configuration, 216, 219
NMM NETCONF data filtering, 239
NMM NETCONF data filtering (conditional match), 243
NMM NETCONF data filtering (regex match), 242
NMM NETCONF message format, 217
NMM NETCONF structure, 216
NMM NETCONF XML view, 220
XML Schema
NMM NETCONF message format, 217